IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 11, No. 9, September, 2024, pp. 2537-2549
Journal homepage: http://iieta.org/journals/mmep

Deep Learning-Based STR Analysis for Missing Person Identification in Mass Casualty Incidents

Donya A. Khalid*, Nasser N. Khamiss

College of Information Engineering, Al-Nahrain University, Baghdad 10011, Iraq

Corresponding Author Email: donya.abbas@nahrainuniv.edu.iq

https://doi.org/10.18280/mmep.110924

Received: 9 April 2023
Revised: 19 June 2023
Accepted: 11 July 2023
Available online: 29 September 2024

Keywords:
DNA, STR, missing person identification, GRU, DNN, Bi-GRU

ABSTRACT

Deoxyribonucleic acid (DNA) profiling is an important branch of forensic science that aids in the identification of missing people, particularly in mass disasters. This study presents an artificial intelligence system that utilizes DNA-Short Tandem Repeat (STR) data to identify victims using Deep Neural Network (DNN), Gated Recurrent Unit (GRU), and Bidirectional GRU (Bi-GRU) deep learning models. The identification of STR information for living family members, such as parents or brothers, poses a significant challenge in victim identification. Familial data are artificially generated based on the actual data of distinct Iraqi individuals from the province of Al-Najaf. Two people are selected as male and female to create a family of 10 members. As a result of this action, 151,580 individuals were generated from 106 different people, which helps to overcome the lack of datasets caused by restrictive policies and the confidentiality of familial datasets in Iraq. These datasets are prepared and formatted for training deep learning models. Based on various reference datasets, the models are built to handle five different scenarios where both parents are alive, only one parent is alive, or the siblings are available for reference. The three models' performances were compared: Bi-GRU performed the best, with a loss of 0.0063 and an accuracy of 0.9979, followed by GRU with a loss of 0.0102 and an accuracy of 0.9964, and DNN with a loss of 0.2276 and an accuracy of 0.9174. The evaluation makes use of a confusion matrix and receiver operating characteristic curve. Based on the literature, this is the first attempt to introduce deep learning in DNA profiling, which reduces both time and effort. Despite the fact that the proposed deep learning models have good results in identifying missing persons according to their families, these models have limitations that can be confined to the availability of familial DNA profiles. The system doesn't work well if no relative samples are available as references, such as a father, mother, or brother. In the future, DNN, GRU, and Bi-GRU models will be applied to mini-STR sequences that are used in cases of degraded victims of incomplete STR sequences.

1. INTRODUCTION

Since its inception in the mid-1980s, the use of forensic DNA profiling has significantly increased [1, 2]. It is well-known as a technique used in forensic investigations for purposes such as identifying criminals from crime scene samples, determining paternity, and identifying the human remains of missing persons. It has become the gold standard for identifying victims, especially in mass-casualty incidents (MCIs) [3, 4].

MCIs can be defined as any sudden and unexpected events that have a negative impact on a community and cause more fatalities than the community is able to handle [5, 6]. There are several potential causes for MCIs, including natural disasters, armed conflicts, terrorist attacks, or accidents. After the Scandinavian Star ferry fire in 1990, DNA analysis was used for the first time to identify mass disaster victims.

For instance, on June 12, 2014, the Islamic State of Iraq and the Levant (ISIL) killed about 1700 members of the Iraqi army and security forces at Camp Speicher in Tikrit, Iraq, resulting in numerous casualties [7].

A forensic examination of human remains discovered in the aftermath of a war or other violent incident has two objectives. First, as part of criminal investigations, it is important to ascertain the cause and manner of death; second, it is important to identify any human remains and, if at all possible, return them to the victim's family. This paper takes the second objective into consideration.

The research proposes deep learning models for missing person identification in mass casualty incidents. rather than manual matching in the traditional approach that manually compares the human profiles of the missing person with those of his or her relatives. In manual matching, only three to four profiles can be compared simultaneously because it takes significant human effort to analyze 15 loci, each locus with two alleles for one short tandem repeat (STR) profile. Introducing an artificially intelligent system for missing person identification minimizes both effort and time

consumption while maximizing the efficiency. The suggested system can also perform the identification regardless to the samples number. Familial dataset generation offers a good chance to create a familial database that may be useful in any other cases of missing person identification.

This paper introduces a deep learning modelling system to assess how closely human DNA profiles of missing persons and samples from living people are similar (mother, father, and brothers). The DNA-short tandem repeat (STR) profile, which consists of 15-loci with two alleles in each locus, is used as input data for this proposed deep learning system. The value that this system outputs indicates whether the missing person is a member of the reference family. The proposed system's strength lies in the development and management of the dataset as well as in the optimal structuring of a deep learning model for the most accurate missing person identification. Figure 1 clarifies a clear overview of the proposed system.

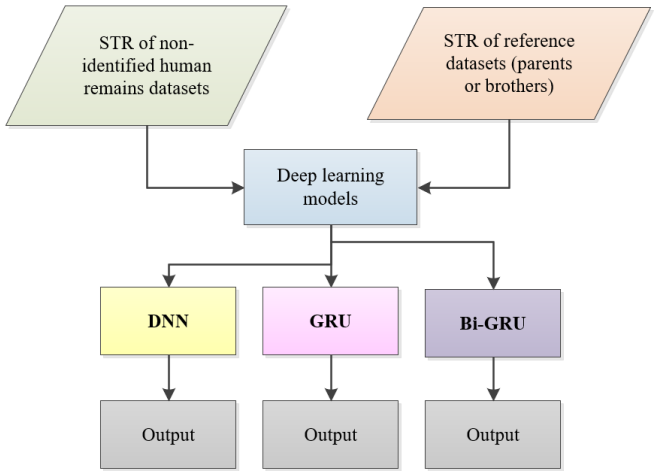


Figure 1. Missing person identification

This paper is arranged as follows: Section 2 gives an overview of missing person identification based on DNA STR. Section 3 presents deep learning in general and gives a description of the three models used. Section 4 describes how the system was modelled and the simulations that were run. Section 5 presents a discussion of the results. Finally, the conclusions are given in Section 6.

2. MISSING PERSON IDENTIFICATION

The following section discusses the increasing demand for DNA analysis in forensic investigations, essentially in cases involving missing persons and disaster victims. The use of DNA profiling, specifically focusing on short tandem repeats (STRs), is highlighted as a crucial technique for identifying individuals. This section also emphasizes the importance of considering broader family relations when direct comparisons are not possible. Furthermore, it provides an overview of the process involved in DNA-STR profiling, including sample collection, amplification, separation, and comparison to reference databases, highlighting its applications in identifying individuals, missing person cases, and paternity testing.

Due to cases involving the identification of missing persons, where DNA profiling and matching are crucial techniques used to identify victims in a disaster, there is an increasing demand for DNA analysis in forensics investigations. The use of DNA analysis was advised by Interpol guidelines, along with dactyloscopy and odontology [8].

For example, in Camp Speicher, according to the Iraqi Ministry of Health, approximately 1153 bodies have been recovered and over 704 victims have been identified, but at the time of writing, 503 corpses are still undergoing verification [7].

Identification is defined as "individualization", the process of comparing DNA samples taken from human remains with living reference family members. However, there is a critical situation that may be difficult to solve, and there is no doubt that such a comparison can be held. For instance, if the victim's parents are deceased or they are very far away from the victim's place, in this situation there is a necessity to define the DNA profile of wider family relations, including brothers, grandfathers, grandmothers, uncles, aunts, cousins, and any other potential living relatives [9].

2.1 DNA–STR

DNA is the molecular blueprint that encodes the genetic information necessary for the growth and operation of all living organisms [10]; these genetic materials are passed down through the generations.

DNA is a long, double-stranded molecule composed of four different types of nitrogenous bases: adenine (A), guanine (G), cytosine (C), and thymine (T) [11]. DNA is useful for determining paternity, finding missing persons, and identifying victims of mass disasters. Therefore, it is essential to try to determine ancestry using STR inference for living relatives [12]. STR is a repeating pattern of two or more nucleotides (the building blocks of DNA) found in a specific region of a chromosome and accounts for approximately 3% of the human genome [13]. STR analysis is a common technique for forensic DNA analysis and personal genotyping; it is used as a genetic marker for human identification. Different genetic markers are used for different purposes in forensic DNA analysis, but STR typing is still the mainstay [14]. This is because of the number of times the pattern is repeated varies from person to person; this variation offers a high degree of discrimination and makes each individual's STR pattern a unique profile [15]. This profile can then be used to identify individuals, either by comparing it to a reference sample (a family member) or by searching it against a database of known profiles.

STRs are inherited like any gene or DNA segment, with each individual having two alleles per STR-one from each parent. An allele refers to one of the alternative forms of a gene that takes a particular locus on a chromosome. In the case of STRs, the alleles are defined by the number of repeats at a particular STR locus.

When an individual inherits alleles for an STR locus, they receive one allele from their biological mother and one allele from their biological father. The combination of alleles determines the individual's genotype for that particular STR locus. For instance, an individual may inherit an allele with 9 repeats from their mother and an allele with 10 repeats from their father, resulting in a genotype of (9, 10) for the D7S820 locus. Paternity can be determined if a child shares about half of a parent's DNA because each parent is responsible for transmitting half of their genetic material [16].

Human profiling and identification based on DNA-STR involve collecting a DNA sample from an individual and then amplifying the STR regions using the polymerase chain

reaction (PCR) method [17]. Then the amplified products are separated based on size using techniques such as capillary electrophoresis [18], after that the resulting STR profile is ready to be compared with a reference database of known DNA profiles in order to identify the individual or to find the likelihood that two DNA samples came from the same person. It is also used in paternity testing and in missing person cases to help identify individuals [19]. Figure 2 shows DNA and STR regions that are found in non-coding regions within DNA.

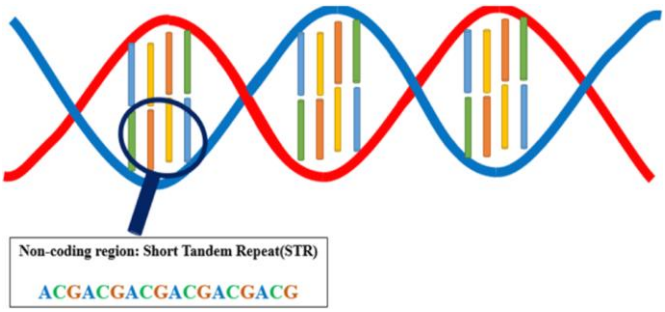


Figure 2. DNA and STR region

3. ARTIFICIAL INTELLIGENCE: DEEP LEARNING

This section gives an overview of deep learning and its application in the field of bioinformatics. Techniques that are used in this work; DNNs, GRUs, and Bi-GRUs are described as well.

Artificial intelligence (AI) is becoming increasingly important in forensic science as it offers the potential to increase the accuracy and effectiveness of various forensic tasks [20]. For example, AI can be used to analyse vast amounts of data in a relatively short length of time, automate tedious tasks, and identify patterns that humans might miss. However, it should be kept in mind that AI must be used in conjunction with human knowledge and judgment [21].

Deep learning is a branch of machine learning and AI that uses neural networks with multiple layers to analyze and model complex data patterns [22, 23].

In deep learning, large datasets are processed by neural networks, enabling them to autonomously learn patterns and relationships. This technique has been successfully applied across various bioinformatics applications to enhance classification accuracy. For instance, many studies have shown that deep learning outperforms traditional methods like random forest or support vector machine algorithms in predicting protein binding and accessibility to DNA sequences [24-28].

Using deep learning in missing persons identification through DNA matching has not been extensively studied. However, there are two relevant studies that apply artificial intelligence techniques to DNA analysis: Anggreainy et al. [29] utilize fuzzy inference for STR-DNA matching across 16 loci, and Siino and Sears [30] employ gradient descent logistic regression for kinship analysis based on 13 loci.

This paper explores and compares three deep learning techniques for an artificial intelligence system that uses DNA-STR data from 15 loci to identify victims: DNNs, GRUs, and bidirectional GRUs (Bi-GRUs). These techniques are detailed in Sections 3.1, 3.2, and 3.3.

3.1 DNNs

DNNs are a type of artificial neural network constructed from multiple layers of artificial neurons, or "units". This approach was inspired by the way the human brain processes information [31].

The neural network has several layers, each of which processes and abstracts information in a unique way. The input data is processed at each layer before being forwarded to the one below it, with the final layer producing the output prediction or classification. When compared to other machine learning techniques, this enables the network to recognize high-level features and representations of the data, improving accuracy and performance [32].

To minimize the error between the predicted output and the true output DNNs are trained by adjusting the weights of the connections between units [33]. This process, known as backpropagation, is repeated multiple times until the error is reduced to an acceptable level. DNNs have proven effective at a variety of tasks, like speech and image recognition, machine translation, and natural language processing. They are particularly proficient at solving problems that call for the discovery of intricate connections between input and output data [34].

However, DNNs can be computationally expensive to train, and the selection of hyperparameters, such as the number of layers and the learning rate, can affect how they perform. They can also be prone to overfitting if the training data is not sufficiently large or diverse.

Each DNN neuron is mathematically represented in the formula below, where the output can be:

$$Z = \sigma(\sum_t x_t . W_t + b) \tag{1}$$

where, x_t is the input vector, W_t is the weight vector, b is bias, Z is the output from the network, and σ is an activation function that may be a leaky rectified linear unit (Leaky ReLU) in all hidden layers or sigmoid in the output layer.

Leaky–ReLU:

$$\sigma_l(x) = \max(ax, x) \tag{2}$$

Sigmoid:

$$\sigma_s(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

3.2 GRUs

GRUs are a kind of RNN, and RNNs are a type of DNN specifically designed for processing sequential data [35]. RNNs are called "recurrent" because they have a loop structure that uses the output from one unit at a one-time step as input for the same unit at the following time step. The importance of processing sequential data lies in capturing temporal dependencies, understanding contextual information, modelling time series patterns, enabling natural language processing, supporting sequential decision-making, predicting future events, and facilitating sequential generation. These applications span a wide range of fields, highlighting the significance of effectively processing and analysing sequential data. RNNs offer these benefits by utilizing recurrent connections and memory mechanisms, allowing them to

process sequential data, capture dependencies, and model temporal information effectively [36].

GRUs were firstly proposed by Cho et al. [37], it enables each recurrent unit to capture dependencies adaptively on different time scales. GRUs uses gating mechanisms in order to control the flow of information between the input, output, and hidden states [38-40].

The memory cell in a GRU is in charge of retaining and losing information over time. The memory cell is made up from two gates, an update gate and a reset gate [41]. Figure 3 depicts a memory cell that is ready for DNA applications.

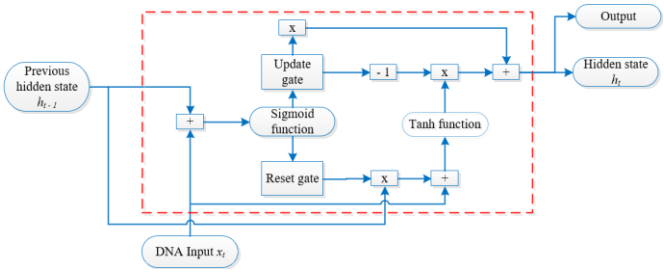


Figure 3. GRU memory cell for DNA identification

A typical GRU unit consists of the following [35, 42]:

Reset gate: The reset gate control how much of the previous hidden state must be forgotten and how much of the current input must be incorporated into the updated hidden state.

$$R = \boldsymbol{\sigma_s}[(W_r \cdot X_t + V_r \cdot h_{(t-1)}) + b_r] \quad (4)$$

where, R is the reset gate at time t; h_{t-1} is the hidden state of the previous time step; X_t is the input at time t; Wr, V_r are the reset gate's learnable weights for input and hidden state of reset gate and b_r is the bias; and $\boldsymbol{\sigma_s}$ is the sigmoid activation function.

After being transformed by the sigmoid function, the new values will all be between 0 and 1, which makes it possible for the gate to distinguish between unimportant and important data.

Update gate: The update gate determines how much of the previous hidden state should be retained and how much of the current input should be incorporated into the updated hidden state. The update gate is calculated next using the same formula as the reset gate; the only difference is the weights of the input vector and hidden state are distinct because each gate has its own set of weights, which implies that the final output vectors of each gate are different. As a result, the gates can be used for their particular purposes.

$$U = \sigma_s[(W_u \cdot x_t + V_u \cdot h_{t-1}) + b_u] \quad (5)$$

where U is the update gate at time t and W_u, V_u are the update gate's learnable weights and b_u is the bias, for the update gate respectively.

Candidate hidden state: The candidate-hidden state is an intermediate value that is computed using the reset gate and the previous hidden state, as well as the current input. This process decides which information is preserved from the prior time steps alongside the new inputs.

$$\tilde{h}_t = \boldsymbol{\sigma_{tanh}}[W_x \cdot x_t + (R \odot (W_h \cdot h_{(t-1)})) + b_h] \quad (6)$$

where, $\tilde{h}_t$ is the candidate hidden state at time t, $\boldsymbol{\sigma_{tanh}}$ is the hyperbolic tangent activation function, and $\odot$ is the Hadamard (element-wise) product operator.

Hidden state: The final hidden state is a weighted combination of the previous hidden state and the candidate's hidden state, as determined by the update gate. If the update gate is close to 1, the candidate hidden state is mostly used to update the hidden state; if it is close to 0, the previous hidden state is mostly retained.

$$h_t = (1-U) \odot h_{(t-1)} + U \odot \tilde{h}_t \quad (7)$$

3.3 Bi-GRUs

Bi-GRUs are another kind of RNN that process input sequences in both forward and backward directions using GRUs [43, 44]. The motivation for using a bidirectional RNN is that the output at each time step may depend on the context from both past and future time steps.

A Bi-GRU processes input sequences using two separate RNNs, known as the forward and backward RNNs, which handle sequences in opposite directions [44]. Each time step's output from both RNNs is concatenated and serves as input for the subsequent step. This setup allows Bi-GRUs to integrate context from both past and future data points in the sequence. Bi-GRUs have proven effective in various applications, such as speech recognition, time series forecasting, and natural language processing. They are particularly valuable in tasks like language modeling and translation, where understanding context from both directions is crucial [45-48].

The Bi-GRU model is calculated based on the state of two GRUs, which are unidirectional in opposite directions. At time step t, the hidden state of the forward GRU is denoted as $\overrightarrow{h_t}$ while the backward GRU at time step t is $\overleftarrow{h_t}$.

$$\overrightarrow{h_t} = \mathrm{GRU}_{\mathrm{forward}}(x_t, \overrightarrow{h_{t-1}}) \quad (8)$$

$$\overleftarrow{h_t} = \mathrm{GRU}_{\mathrm{backward}}(x_t, \overleftarrow{h_{t+1}}) \quad (9)$$

At each time step, the final result is obtained by concatenating the forward and backward hidden states:

$$y_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \quad (10)$$

where, $\oplus$ indicates the process of concatenating two vectors.

4. SYSTEM MODELLING AND SIMULATION

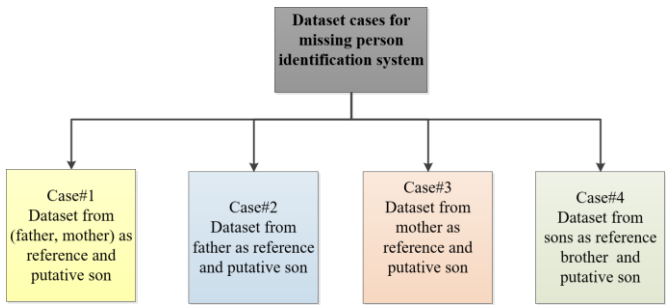


Figure 4. Proposed dataset scenarios

The proposed system utilizes three deep learning models to identify human remains or missing persons. Due to the

absence of familial datasets for deceased individuals, it introduces a method to generate artificial STRs for family members, creating a synthetic familial dataset. This dataset is then manipulated to simulate various identification scenarios. The deep learning models are applied to implement human identification tasks based on these scenarios. Details of these dataset scenarios are illustrated in Figure 4.

4.1 Data creation

The first step in this system is data preparation, which verifies the missing identification of human remains. For the purpose of matching the deceased son's alleles with his putative reference family, 15 loci of STR are used. Table 1 shows the applied STR-15 loci.

The proposed data creation model (Figure 5) uses 106 real, unrelated Iraqi individuals from the Najaf province of Iraq [49] and creates 53 familial datasets by using two individuals as male and female parents. The familial dataset structure consists of mother, father, and 10 created children, 5 of whom belong to the family, and 5 of whom do not, according to the 530 data samples. The purpose of this step is to make the proposed system knows the relationship between the parents and their corresponding children.

Each family set has a unique number range from 0 to 52. All family datasets also have labels appended, with a label (1) indicating whether the son actually belongs to the family in question or not (0). The resulting datasets for one family are shown in Table 2, for instance the table shows the 10th family member set, like any family set it is consisting form father, mother, five correct sons, and five incorrect sons. Each of the 15 STR loci has two alleles (A1 and A2).

Table 1. Applied STR-15 loci [49]

D8S1179	D21S11	D7S820	CSF1PO	D3S1358	TH01	D13S317	D16S539	D2S1338	D19S433	vWA	TPOX	D18S51	D5S818	FGA

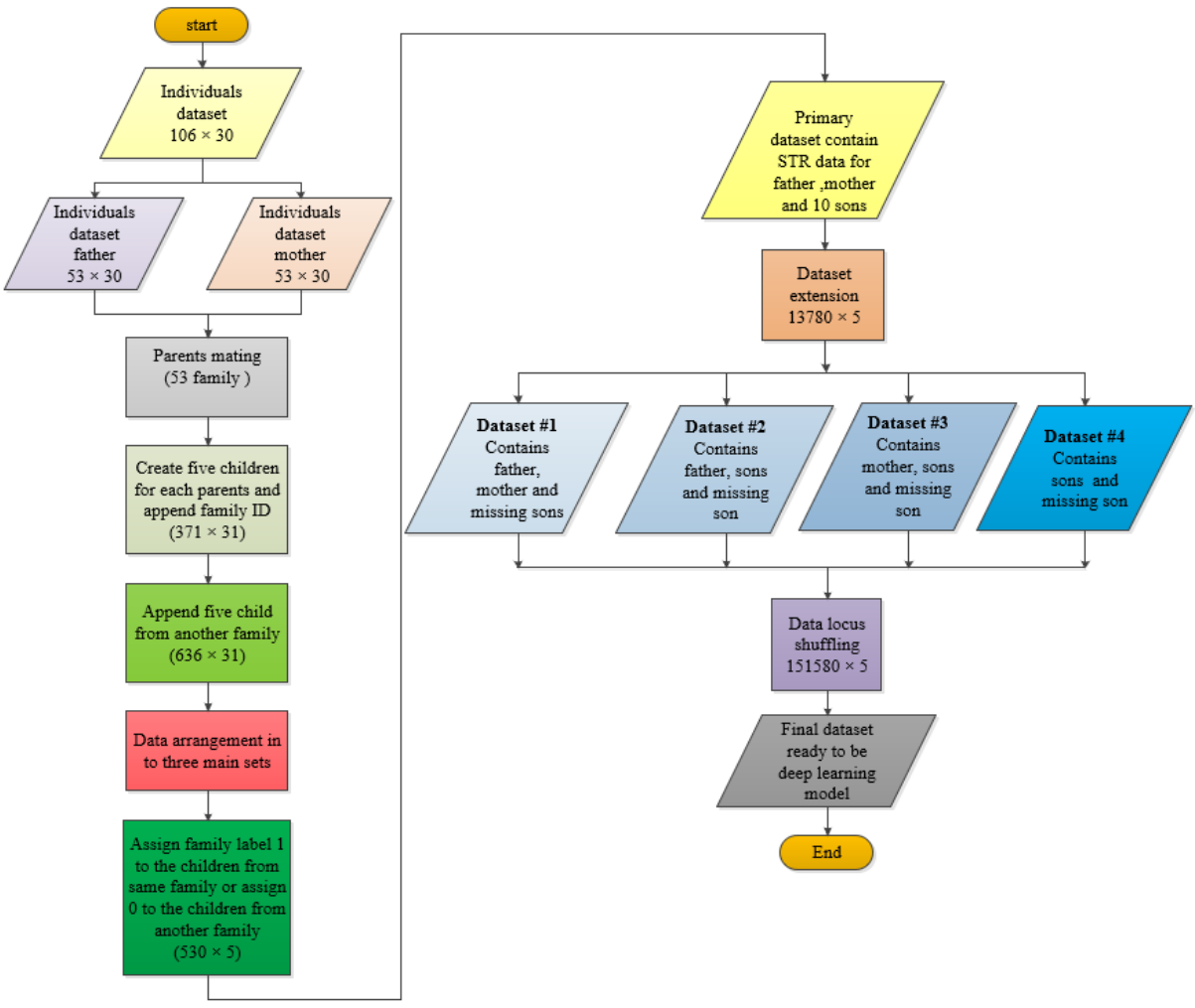


Figure 5. Data creation steps

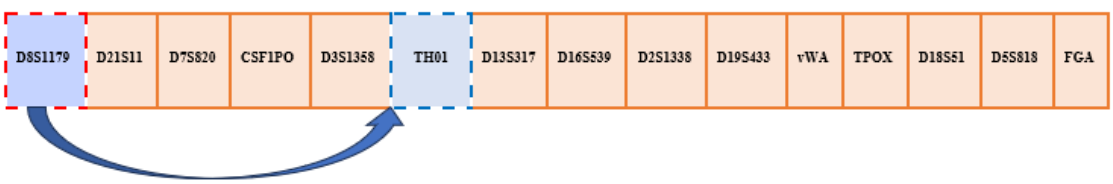


Figure 6. Locus shuffling

Table 2. Dataset of one family of 53

Father Set	Mother Set	Child Set	Family ID	Label
Father 1	Mother 1	Child 1	0	1
Father 1	Mother 1	Child 2	0	1
Father 1	Mother 1	Child 3	0	1
Father 1	Mother 1	Child 4	0	1
Father 1	Mother 1	Child 5	0	1
Father 1	Mother 1	Child 6	0	0
Father 1	Mother 1	Child 7	0	0
Father 1	Mother 1	Child 8	0	0
Father 1	Mother 1	Child 9	0	0
Father 1	Mother 1	Child 10	0	0

Table 3. Suggested dataset

No.	Set 1	Set 2	Target
1	Father	Mother	Missing Son
	Mother	Father	Missing Son
2	Mother	Reference Son	Missing Son
	Reference Son	Mother	Missing Son
3	Father	Reference Son	Missing Son
	Reference Son	Father	Missing Son
4	Son	Reference Son	Missing Son

The system also introduces a method for randomly shuffling these 15 loci of STR sequences, which is shown in Figure 6.

The locus position sequencing is maintained across all family members during shuffling, so if the D8S1179 locus alleles values on the father dataset is moved to a specific position, the same locus values will be moved in a similar fashion on the mother and child datasets, and so on. The importance of this process is to train the three deep-learning models independently on the values of each locus by applying the shuffling procedure to the parents of 53 families. The dataset size will be 53,530 samples after 11 iterations of shuffling.

The created datasets are managed in such a manner as to solve four cases, as presented in Table 3. The resultant dataset has a size equal to 151,580 samples.

4.2 Proposed missing DNA deep learning systems

Three deep learning models have been suggested in order to determine the identity of a missing person. These models, which are illustrated in Figure 5 and outlined in Section 3, consist of the DNN, GRU, and Bi-GRU. The rationale for choosing DNN, GRU, and Bi-GRU models lies in their respective strengths and suitability for the analysis of DNA-STR data in the context of missing person identification. DNNs excel in capturing complex patterns; GRUs are effective in handling sequential data and long-term dependencies; and Bi-GRUs leverage bidirectional processing for a comprehensive understanding of DNA-STR relationships, all of which are advantageous for accurate identification in mass casualty incidents.

One can deploy the optimal model as a web API by integrating it within a web framework like Flask or Django. This involves creating an API endpoint that processes input data and delivers predictions. The API can be hosted on a web server or a cloud platform to facilitate access over the internet. It is crucial to implement appropriate authentication and security measures to control access and ensure the reliable availability of the API for making predictions using the Bi-GRU model. The following sections will detail the architectural design for each model.

4.2.1 Missing DNA: DNN system design

The proposed DNN-identification system (Figure 7) is constructed from embedded layer, flatten layer, dense layer and dropout layer. This model takes as its input the datasets for the deceased missing person and the corresponding living reference person. The output of this model is a decision as to whether or not the missing person belongs to the same family as the reference person.

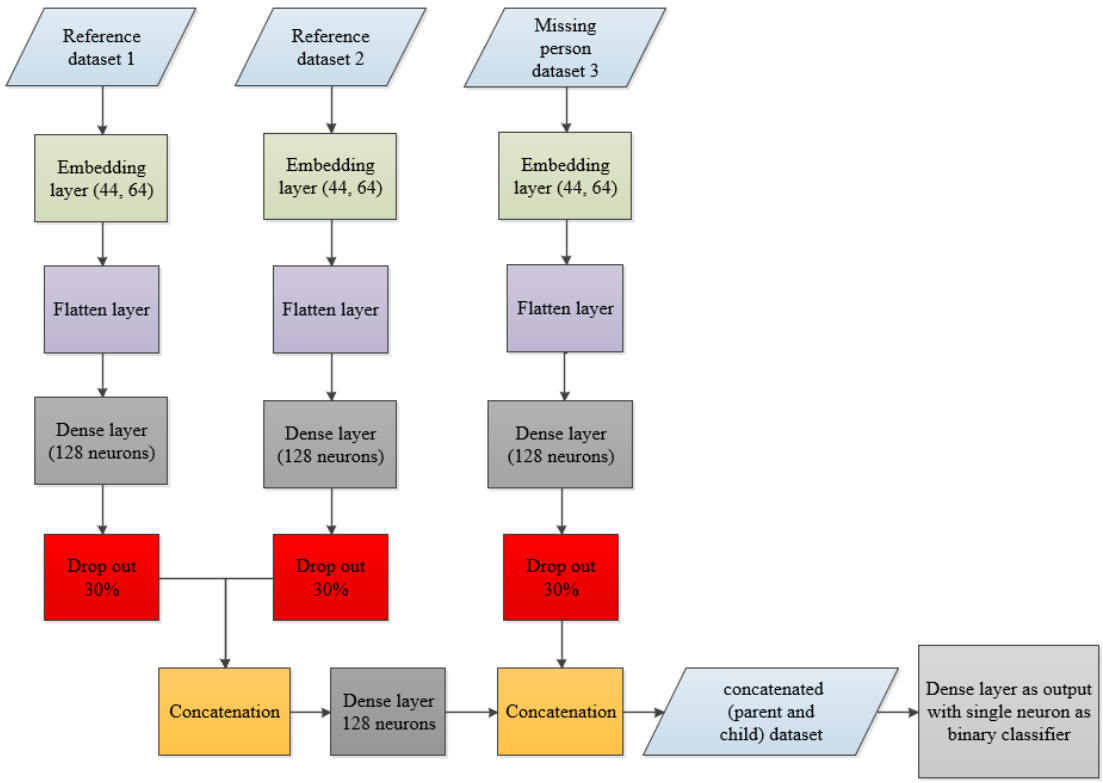


Figure 7. DNN model structure

Table 4. DNN parameters

DNA Model: DNN		
Layer (Shape)	**Output Shape**	**Parameters**
embedding (Embedding)	multiple	2816
Dense (Dense)	multiple	245,888
Dropout (Dropout)	multiple	0
dense_1 (Dense)	multiple	245,888
dropout_1 (Dropout)	multiple	0
dense_2 (Dense)	multiple	32,896
dropout_2 (Dropout)	multiple	0
dense_3 (Dense)	multiple	245,888
dense_4 (Dense)	multiple	32,896
dense_4 (Dense)	multiple	129
Total parameters: 806,401		
Trainable parameters: 806,401		
Non-trainable parameters: 0		

The model has 8 layers: an embedding layer, five dense layers, and three dropout layers, the total trainable parameters in the model is 806,401, and there are no non-trainable parameters. Table 4 summarizes the DNA-DNN model parameters.

4.2.2 Missing DNA: GRU system design

The proposed structure for the GRU model (Figure 8) is constructed from embedded layer, GRU layer and dropout layer. This model takes as input the datasets for the missing person and the corresponding living reference person. The output of this model is a decision as to whether or not the missing person belongs to the same family as the reference person.

Table 5 gives the description of DNA-GRU model parameters. It has seven layers: an embedding layer, two dense layers, five GRU layers, and a dropout layer. The total number of model parameters equals to 157,377, all of which are trainable parameters.

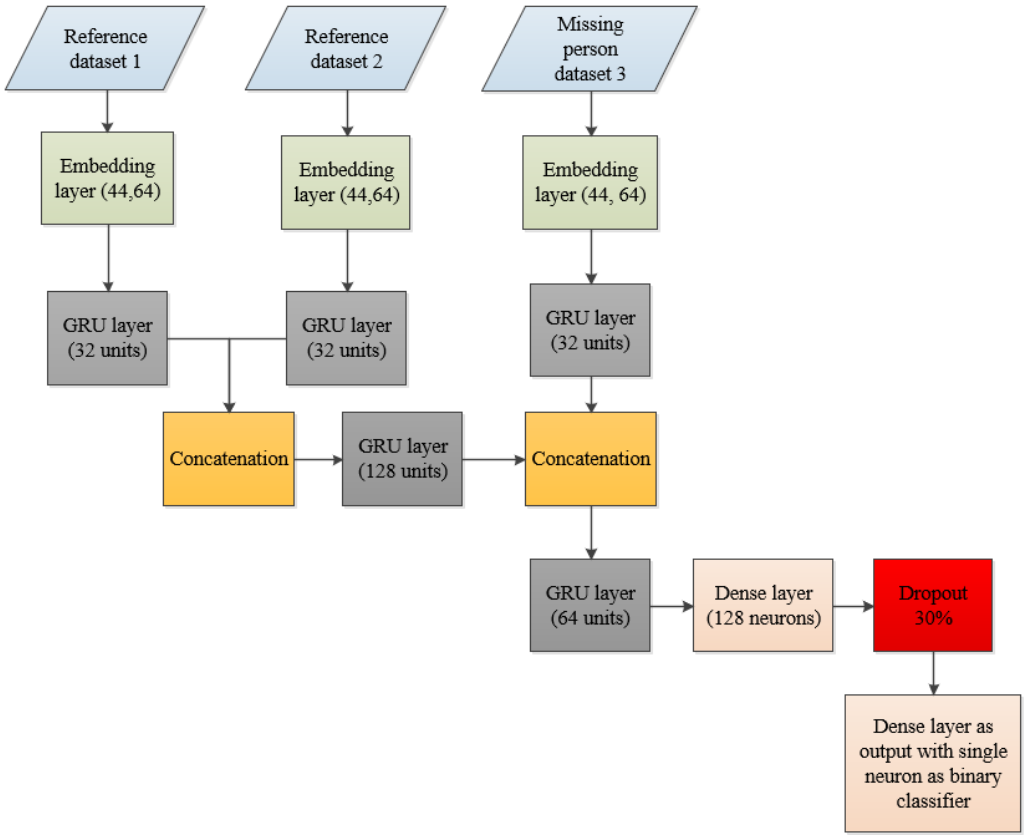


Figure 8. GRU model structure

Table 5. GRU parameters

DNA Model: GRU		
Layer (Shape)	**Output Shape**	**Parameters**
embedding (Embedding)	multiple	2816
Gru (GRU)	multiple	9408
gru_1 (GRU)	multiple	9408
gru_2 (GRU)	multiple	74,496
gru_3 (GRU)	multiple	9408
gru_4 (GRU)	multiple	43,392
Dense (Dense)	multiple	8320
Dropout t(Dropout)	multiple	0
dense_1 (Dense)	multiple	120
Total parameters: 157,377		
Trainable parameters: 157,377		
Non-trainable parameters: 0		

4.2.3 Missing DNA: BI-GRU system design

The third model proposed is a Bi-GRU, which is constructed with an embedded layer, a Bi-GRU layer, and a dropout layer. Figure 9 illustrates the structure of the proposed Bi-GRU identification system.

The DNA-Bi-GRU model parameters are presented in Table 6. This model has an embedding layer, five bidirectional GRU layers, two dense layers, single GRU layer, and a dropout layer. The model has a total of 422,273 parameters, all of which are trainable.

No previous studies employed the same target of identifying missing persons using deep learning techniques, but research [29] is selected as the baseline model since it has the idea of DNA matching using artificial intelligence. This baseline model develops an algorithm for implementing fuzzy

similarity to evaluate the degree of similarity between human DNA profiles. The innovation of our research in this paper is highlighted by the fact that the deep learning models yield optimal results compared to the baseline model [29], and these results are achieved on the largest datasets with a size equal to 1515580 samples, while [29] applied to only 100 samples.

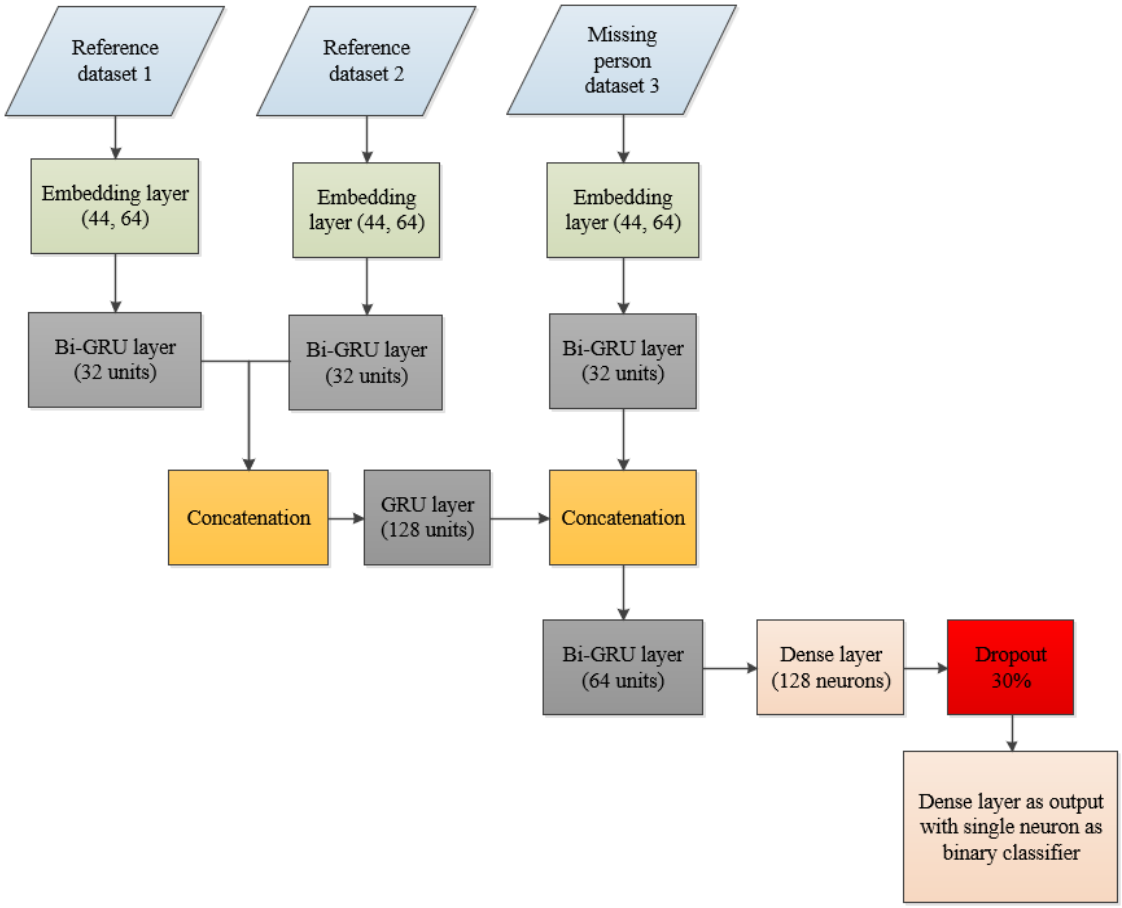


Figure 9. Bi-GRU model structure

Table 6. Bi-GRU parameters

DNA Model: Bi-GRU		
Layer (Shape)	**Output Shape**	**Parameters**
embedding (Embedding)	multiple	2816
Bidirectional (Bidirectional)	multiple	18,816
bidirectional_1 (Bidirectional)	multiple	18,816
bidirectional_2 (Bidirectional)	multiple	198,144
bidirectional_3 (Bidirectional)	multiple	18,816
bidirectional_4 (Bidirectional)	multiple	148,224
dense (Dense)	multiple	16,512
dropout (Dropout)	multiple	0
dense_1 (Dense)	multiple	129
Total parameters: 422,273		
Trainable parameters: 422,273		
Non-trainable parameters: 0		

4.3 Performance evaluation metrics

To compare and analyse the efficiency of the proposed deep learning model's different metrics are used in this research, they are:

Accuracy: The proportion of all true predictions to all predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

Sensitivity (Recall): It is the ratio of true positives to the total actual positives.

$$Sensitivity = \frac{TP}{TP + FN} \tag{12}$$

Precision: It is the ratio of true positives to the total predicted positives.

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

F1-Score: It is the harmonic mean of precision and sensitivity. It used to measure the performance of the test for the positive class.

$$F1 - Score = 2 \cdot \frac{Sensitivity \cdot precision}{Sensitivity + precision} \tag{14}$$

Receiver operating characteristic (ROC) curve: A graph illustrating the performance of a classification model across all thresholds. The ROC curve displays two metrics: the true positive rate and the false positive rate. The Area Under Curve (AUC) serves as a summary metric for the ROC curve, indicating the effectiveness of a binary classifier in distinguishing between classes.

The predictions are evaluated based on four metric types: True positive (TP), false positive (FP), true negative (TN), and false negative (FN).

5. RESULTS AND DISCUSSION

The results were carried out based on Google-Colab, which is based on the Python programming language. The necessary libraries, including TensorFlow, Numpy, Pandas, Matplotlib, Scikit-learn, Seaborn, and Keras, were imported. The subsequent sections detail the partitioning of the dataset into training and testing sets, as well as the performance evaluations of the three proposed identification systems.

5.1 Training/testing dataset

During the training and testing phases, the entire dataset (151,580) was divided into 80% training and 20% testing sets. The training dataset was further divided into balanced labels, with half of the data labelled as 1 (indicating true children) and the other half labelled as 0 (indicating false children). Figure 10 shows the distribution of labels in a histogram. The training and testing processes were carried out separately for each of the proposed DNA deep learning models in order to make comparisons and evaluations.

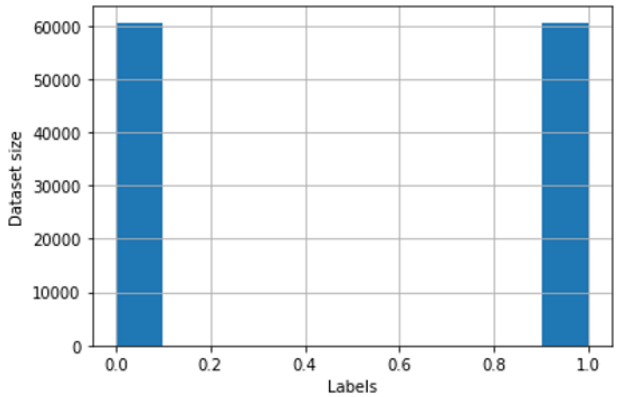
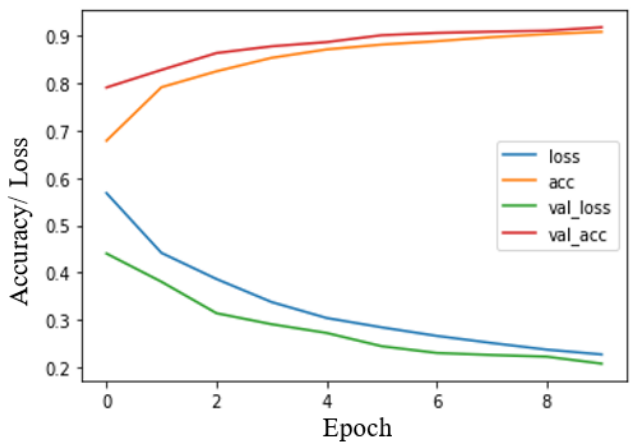


Figure 10. Histogram of the training dataset

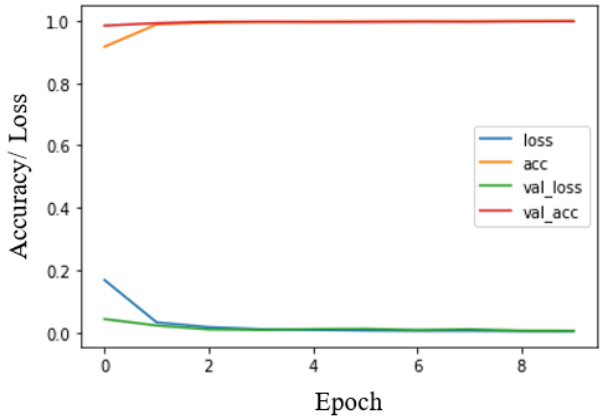
5.2 Performance comparison of the three classifiers

Applying deep learning models such as DNN, GRU, and Bi-GRU on DNA-STR datasets yields varying results in identifying missing persons.

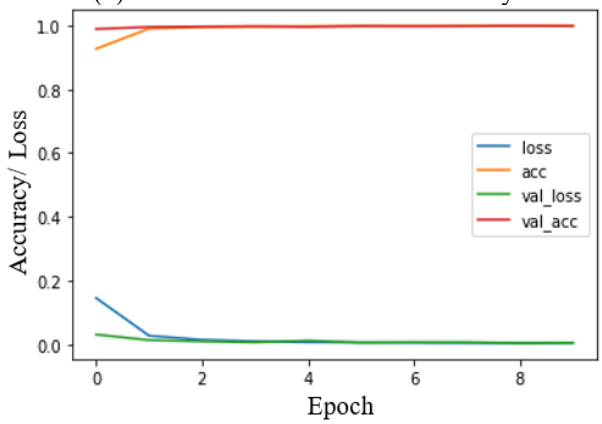
The next step is the presentation of experimental results that support the work's primary goal, which is a binary classification for missing person identification in mass casualty incidents (MCI), the term MCI will be appended in each type of model results.



(a) MCI-DNN model loss and accuracy



(b) MCI-GRU model loss and accuracy



(c) MCI-Bi-GRU model loss and accuracy

Figure 11. Performance of loss and accuracy for the three models

Accuracy results

In Figures 11 (a)-(c) accuracy and loss for DNN, GRU and Bi-GRU are presented respectively.

When comparing the performance of the three models over 10 epochs in Figure 11, it can be deduced that Bi-GRU and GRU models give the best performance in terms of accuracy and loss, while DNN has a higher loss and a lower accuracy value.

Confusion matrix results

Also, to compare the performance of the three models, the confusion matrix is also shown in Figures 12 (a)-(c) for the three models DNN, GRU, and Bi-GRU, respectively.

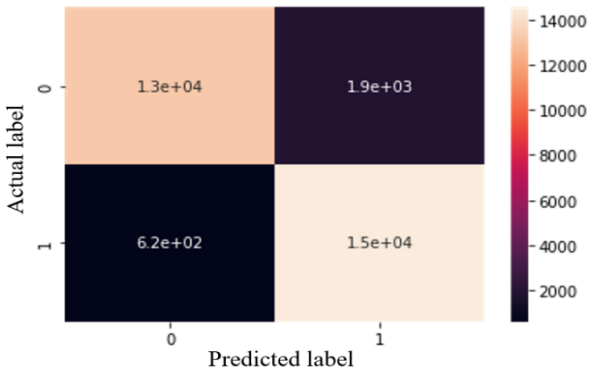
From the illustration of the confusion matrix, one can conclude that the Bi-GRU model gives the minimum total incorrect predictions (FP+FN) equal to 45, followed by GRU, which gives 53, and DNN, which gives the highest number of incorrect predictions equal to 2520. The TP and TN values are mostly equal due to equally distributed tested datasets.

ROC results

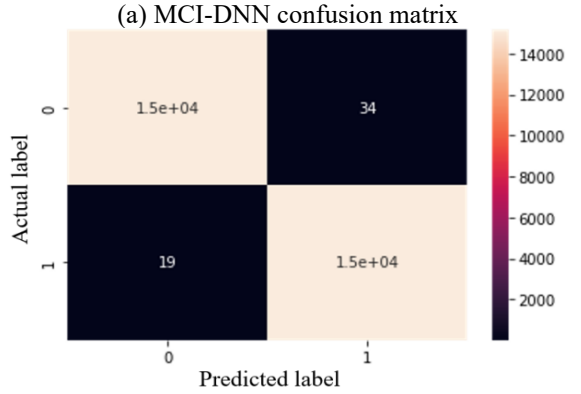
To support a more exhaustive analysis of the three binary classifier systems' performance, ROC is illustrated in Figures 13 (a-c) for DNN, GRU, and Bi-GRU, respectively.

The ROC curve is used to assess system performance as a binary classifier and demonstrates strong results, as seen in Figure 13. It visualizes the trade-off between the true positive rate (TPR) and the false positive rate (FPR), with a perfect area under the curve equal to 0.999 for the Bi-GRU model, followed by 0.998 for GRU, and ending with 0.917 at DNN.

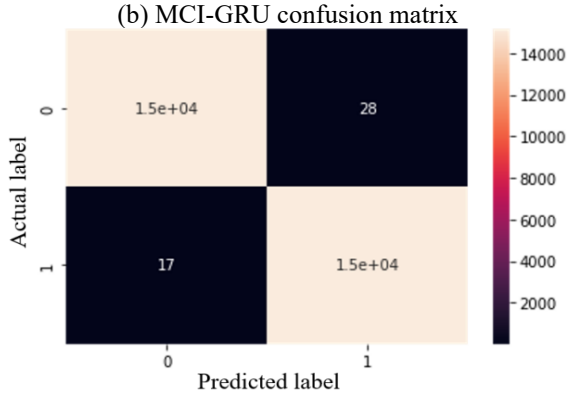
Tables 7 and 8 display a performance comparison of the three proposed models in terms of accuracy, loss, validation accuracy, and validation loss.



(a) MCI-DNN confusion matrix



(b) MCI-GRU confusion matrix

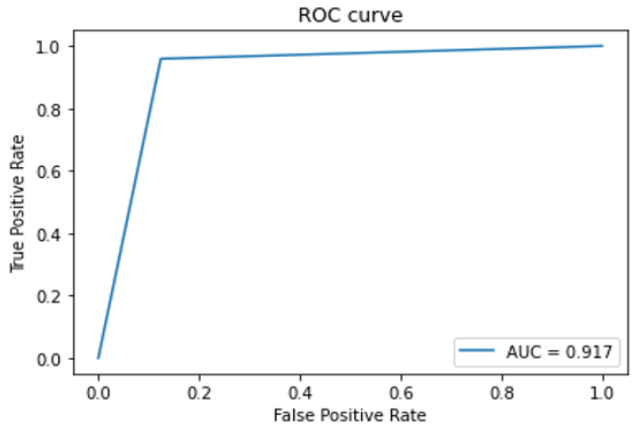


(c) MCI-GRU confusion matrix

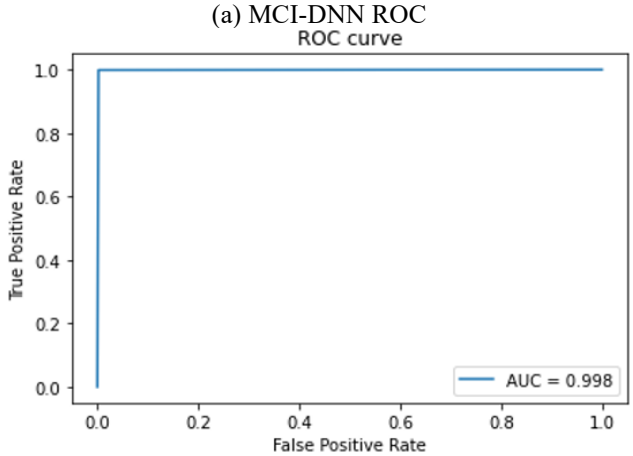
Figure 12. Confusion matrix for the three models

Based on Table 7 provided, one can observe that the training and validation losses for the three models DNN, GRU, and Bi-GRU, trained for ten epochs. As the number of epochs increases, the training and validation losses for all models decrease. However, it can be observed that the Bi-GRU model achieves the lowest training and validation losses comparing with DNN and GRU models.

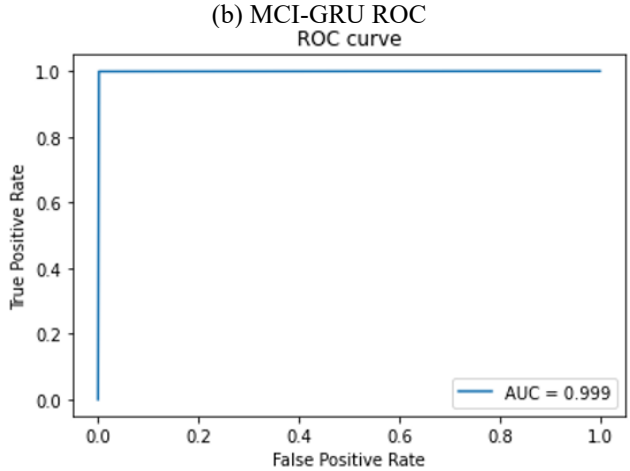
In terms of the DNN model, it can be seen that the validation loss decreases slower compared to the other models, indicating that the model may not be generalizing well to the validation data.



(a) MCI-DNN ROC



(b) MCI-GRU ROC



(c) MCI-Bi-GRU ROC

Figure 13. ROC curve for the three models

Table 7. Comparison between loss and validation loss for the three models

Models	DNN		GRU		Bi-GRU	
Epochs	**Loss**	**Validation Loss**	**Loss**	**Validation Loss**	**Loss**	**Validation Loss**
1	0.5678	0.4404	0.1681	0.0428	0.1455	0.0310
2	0.4417	0.3808	0.0323	0.0221	0.0273	0.0134
3	0.3866	0.3145	0.0168	0.0095	0.0144	0.0095
4	0.3381	0.2913	0.0102	0.0081	0.0099	0.0064
5	0.3044	0.2729	0.0079	0.0107	0.0068	0.0121
6	0.2847	0.2449	0.0059	0.0114	0.0063	0.0056
7	0.2667	0.2307	0.0050	0.0077	0.0054	0.0071
8	0.2517	0.2264	0.0051	0.0101	0.0045	0.0072
9	0.2377	0.2229	0.0043	0.0058	0.0044	0.0045
10	0.2276	0.2082	0.0040	0.0053	0.0047	0.0053

Table 8. Comparison between accuracy and validation accuracy for the tree models

Models	DNN		GRU		Bi-GRU	
Epochs	Accuracy	Validation Accuracy	Accuracy	Validation Accuracy	Accuracy	Validation Accuracy
1	0.6782	0.7904	0.9168	0.9844	0.9269	0.9889
2	0.7910	0.8275	0.9884	0.9922	0.9902	0.9953
3	0.8245	0.8630	0.9942	0.9966	0.9949	0.9966
4	0.8529	0.8772	0.9964	0.9972	0.9966	0.9974
5	0.8707	0.8861	0.9974	0.9962	0.9978	0.9958
6	0.8809	0.9006	0.9980	0.9964	0.9979	0.9981
7	0.8880	0.9055	0.9983	0.9970	0.9982	0.9974
8	0.8966	0.9082	0.9982	0.9968	0.9984	0.9978
9	0.9030	0.9101	0.9986	0.9977	0.9985	0.9985
10	0.9079	0.9174	0.9986	0.9983	0.9983	0.9981

Table 9. Classification report

Metrics\Models	DNN	GRU	Bi-GRU
Precision	0.92	1.0	1.0
Recall	0.92	1.0	1.0
F1-Score	0.92	1.0	1.0

Table 10. Models comparison with other related artificial intelligence works

Research	Number of Profiles	Accuracy
Baseline fuzzy model [29]	100	80%
Proposed DNN	151,580	91.7%
Proposed GRU	151,580	99.6%
Proposed Bi-GRU	151,580	99.7%

Precision, recall, and F1-score results

Table 9 appears to be a classification report showing the precision, recall, and F1-score for binary class (0 and 1) in a dataset of 30,316 instances. The metrics are compared among the three models (DNN, GRU, and Bi-GRU), and the DNN model has the lowest scores of the three models.

The DNN model appears to be the least effective of the three models, as it consistently achieves lower accuracy and validation accuracy scores compared to the GRU and Bi-GRU models, in which GRU and Bi-GRU are more effective in capturing the temporal dependencies and patterns present in the STR sequences, The GRU and Bi-GRU models outperform the DNN model because of their recurrent connections and memory mechanisms, which let them remember and apply data from earlier time steps. This feature is especially useful when processing sequential data, such as STR sequences, where the order and context of the data elements are critical for precise analysis. It's worth noting that the GRU model performs well, but not as well as the Bi-GRU model. This is likely because the Bi-GRU can process input sequences in both forward and backward directions, enabling it to capture more contextual information from the input data.

The average validation loss for the DNN model was found to be 0.2833, while the GRU model exhibited a significantly lower average validation loss of 0.01335 and the Bi-GRU model had the lowest average validation loss of 0.01021. A statistical analysis using a t-test revealed a highly significant difference (t-value = 11.0969, $p < 0.001$) between the mean validation losses of the DNN and GRU models, indicating that the GRU model outperformed the DNN model in terms of validation loss. Similarly, the Bi-GRU model also demonstrated superior performance compared to the DNN model. These findings suggest that the GRU and Bi-GRU models exhibit better predictive capability and are more effective in minimizing the validation loss compared to the DNN model.

Table 10 compares the proposed models with the baseline model regarding the number of profiles used and reported accuracy. It indicates that the proposed MCI-Bi-GRU model achieves the highest accuracy, utilizing 151,580 sample profiles.

To calculate the percentage of improvement, the accuracy of each proposed model can be compared with the baseline model [29]. It can be concluded that Bi-GRU improvement over Fuzzy is equal to 24.74%, GRU improvement over Fuzzy is equal to 24.55%, and DNN improvement over Fuzzy is equal to 14.68%.

6. CONCLUSION

The objective of this paper is to propose a method for identifying missing persons using DNA-STR and to evaluate the performance of three deep learning models (DNN, GRU, and Bi-GRU) in this task. The models are binary classifiers that aim to determine the correct family of the missing person by comparing their STR profile with the biological reference family.

The results demonstrate that the GRU and Bi-GRU models show a significant improvement in the validation loss and accuracy compared to the DNN model, and the Bi-GRU model performs the best among all models, achieving the lowest loss. To calculate the percentage of improvement, one can compare the accuracy of each model with the lowest-performing model (DNN), Bi-GRU improvement over DNN is equal to 8.77%, while GRU improvement over DNN is equal to 8.60%.

Overall, it can be concluded that the Bi-GRU model is the best choice for this task among other proposed models based on the provided metrics (accuracy, loss, confusion matrix, and AUC). It gives the highest accuracy equal to 0.99, a minimum loss equal to 0.004, a minimum number of false predictions equal to 45, an AUC equal to 0.999, and a precision, recall, and f1-score equal to 1.00. This is due to the fact that at each time step the Bi-GRU model uses inputs from both forward and backward directions, which leading to better performance.

Despite the fact that the proposed deep learning models have good results in identifying missing persons according to their families, these models have limitations that can be confined to the availability of familial DNA profiles. The system doesn't work well if no relative samples are available as references, such as a father, mother, or brother.

The proposed Bi-GRU model can be deployed using a web API and involves wrapping the trained Bi-GRU model in a web framework to create an API endpoint to be used in real-world applications for DNA profiling. In the future, DNN,

GRU, and Bi-GRU models will be applied to mini-STR sequences that are used in cases of degraded victims of incomplete STR sequences.

REFERENCES

[1] Butler, J.M. (2015). The future of forensic DNA analysis. Philosophical Transactions of the Royal Society B: Biological Sciences, 370(1674): 20140252. https://doi.org/10.1098/rstb.2014.0252

[2] Gill, P., Jeffreys, A.J., Werrett, D.J. (1985). Forensic application of DNA 'fingerprints'. Nature, 318(6046): 577-579. https://doi.org/10.1038/318577a0

[3] Chatfield, L. (2002). Forensic DNA typing: Biology and technology behind STR markers. Heredity, 89(4): 327-327. https://doi.org/10.1038/sj.hdy.6800124

[4] Clayton, T.M., Whitaker, J.P., Maguire, C.N. (1995). Identification of bodies from the scene of a mass disaster using DNA amplification of short tandem repeat (STR) loci. Forensic Science International, 76(1): 7-15. https://doi.org/10.1016/0379-0738(95)01787-9

[5] Brough, A.L., Morgan, B., Rutty, G.N. (2015). The basics of disaster victim identification. Journal of Forensic Radiology and Imaging, 3(1): 29-37. https://doi.org/10.1016/j.jofri.2015.01.002

[6] Montelius, K., Lindblom, B. (2012). DNA analysis in disaster victim identification. Forensic Science, Medicine, and Pathology, 8(2): 140-147. https://doi.org/10.1007/s12024-011-9276-z

[7] Salihy, R. (2019). Terror and torment: The civilian journey to escape Iraq's war against the "Islamic state". In Iraq After ISIS: The Challenges of Post-War Recovery, pp. 79-98. https://doi.org/10.1007/978-3-030-00955-7_6

[8] INTERPOL (2023). Disaster victim identification guide. https://www.interpol.int/How-we-work/Forensics/Disaster-Victim-Identification-DVI.

[9] Vigeland, M.D., Marsico, F.L., Pinero, M.H., Egeland, T. (2020). Prioritising family members for genotyping in missing person cases: A general approach combining the statistical power of exclusion and inclusion. Forensic Science International: Genetics, 49: 102376. https://doi.org/10.1016/j.fsigen.2020.102376

[10] Watson, J.D., Crick, F.H.C. (1993). Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. JAMA, 269(15): 1966-1967. https://doi.org/10.1001/jama.1993.03500150078030

[11] International Human Genome Sequencing Consortium. (2001). Initial sequencing and analysis of the human genome. Nature, 409(6822): 860-921. https://doi.org/10.1038/35057062

[12] Butler, J.M. (2014). Advanced Topics in Forensic DNA Typing: Interpretation. Academic Press.

[13] Wyner, N., Barash, M., McNevin, D. (2020). Forensic autosomal short tandem repeats and their potential association with phenotype. Frontiers in Genetics, 11: 884. https://doi.org/10.3389/fgene.2020.00884

[14] Keerti, A., Ninave, S. (2022). DNA fingerprinting: Use of autosomal short tandem repeats in forensic DNA typing. Cureus, 14(10): e30210. https://doi.org/10.7759/cureus.30210

[15] Niedzwiecki, E., Debus-Sherrill, S., Field, M.B. (2017). Understanding familial DNA searching: Coming to a consensus on terminology. Fairfax.

[16] Lamb, M.E., Sutton-Smith, B., Sutton-Smith, B., Lamb, M.E. (Eds.). (2014). Sibling Relationships: Their Nature and Significance Across the Lifespan. Psychology Press.

[17] Mullis, K.B., Faloona, F.A. (1987). Specific synthesis of DNA in vitro via a polymerase-catalyzed chain reaction. Methods in Enzymology, 155: 335-350. https://doi.org/10.1016/0076-6879(87)55023-6

[18] Heller, C. (2001). Principles of DNA separation with capillary electrophoresis. Electrophoresis, 22(4): 629-643. https://doi.org/10.1002/1522-2683(200102)22:4<629::AID-ELPS629>3.0.CO;2-S

[19] Norrgard, K. (2008). Forensics, DNA fingerprinting, and CODIS. Nature Education, 1(1): 35.

[20] LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. Nature, 521(7553): 436-444. https://doi.org/10.1038/nature14539

[21] Mirza, M., Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784. https://doi.org/10.48550/arXiv.1411.1784

[22] Hung, C.L., Tang, C.Y. (2017). Bioinformatics tools with deep learning based on GPU. In 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Kansas City, MO, USA, pp. 1906-1908, pp. 1906-1908. https://doi.org/10.1109/BIBM.2017.8217950

[23] Zhou, J., Troyanskaya, O.G. (2015). Predicting effects of noncoding variants with deep learning–based sequence model. Nature Methods, 12(10): 931-934. https://doi.org/10.1038/nmeth.3547

[24] Alipanahi, B., Delong, A., Weirauch, M.T., Frey, B.J. (2015). Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. Nature Biotechnology, 33(8): 831-838. https://doi.org/10.1038/nbt.3300

[25] Kopp, W., Monti, R., Tamburrini, A., Ohler, U., Akalin, A. (2020). Deep learning for genomics using Janggu. Nature Communications, 11(1): 3488. https://doi.org/10.1038/s41467-020-17155-y

[26] Zou, J., Huss, M., Abid, A., Mohammadi, P., Torkamani, A., Telenti, A. (2019). A primer on deep learning in genomics. Nature Genetics, 51(1): 12-18. https://doi.org/10.1038/s41588-018-0295-5

[27] Alharbi, W.S., Rashid, M. (2022). A review of deep learning applications in human genomics using next-generation sequencing data. Human Genomics, 16(1): 26. https://doi.org/10.1186/s40246-022-00396-x

[28] Telenti, A., Lippert, C., Chang, P.C., DePristo, M. (2018). Deep learning of genomic variation and regulatory network data. Human Molecular Genetics, 27(Supplement_R1): R63-R71. https://doi.org/10.1093/hmg/ddy115

[29] Anggreainy, M.S., Widyanto, M.R., Widjaja, B., Soedarsono, N., Widodo, P.T. (2019). Family relation and STR-DNA matching using fuzzy inference. International Journal of Electrical & Computer Engineering, 9(2): 1335-1345. https://doi.org/10.11591/ijece.v9i2.pp1335-1345

[30] Siino, V., Sears, C. (2020). Artificially intelligent scoring and classification engine for forensic identification. Forensic Science International: Genetics, 44: 102162. https://doi.org/10.1016/j.fsigen.2019.102162

[31] Miikkulainen, R., Liang, J., Meyerson, et al. (2024). Evolving deep neural networks. In Artificial Intelligence

in the Age of Neural Networks and Brain Computing, pp. 269-287. https://doi.org/10.1016/B978-0-12-815480-9.00015-3

[32] Tang, B., Pan, Z., Yin, K., Khateeb, A. (2019). Recent advances of deep learning in bioinformatics and computational biology. Frontiers in Genetics, 10: 214. https://doi.org/10.3389/fgene.2019.00214

[33] Samek, W., Binder, A., Montavon, G., Lapuschkin, S., Müller, K.R. (2016). Evaluating the visualization of what a deep neural network has learned. IEEE Transactions on Neural Networks and Learning Systems, 28(11): 2660-2673. https://doi.org/10.1109/TNNLS.2016.2599820

[34] Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. Neurocomputing, 234: 11-26. https://doi.org/10.1016/j.neucom.2016.12.038

[35] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014) Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, pp. 1724-1734. https://doi.org/10.3115/v1/D14-1179

[36] Salehinejad, H., Sankar, S., Barfett, J., Colak, E., Valaee, S. (2017). Recent advances in recurrent neural networks. arXiv preprint arXiv:1801.01078. https://doi.org/10.48550/arXiv.1801.01078

[37] Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y. (2014) On the properties of neural machine translation: Encoder–decoder approaches. In Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, pp. 103-111. https://doi.org/10.3115/v1/W14-4012

[38] Chung, J., Gulcehre, C., Cho, K.H., Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555. https://doi.org/10.48550/arXiv.1412.3555

[39] Chaudhari, S., Mithal, V., Polatkan, G., Ramanath, R. (2021). An attentive survey of attention models. ACM Transactions on Intelligent Systems and Technology, 12(5): 53. https://doi.org/10.1145/3465055

[40] Tang, D., Qin, B., Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, pp. 1422-1432. https://doi.org/10.18653/v1/D15-1167

[41] Yao, K., Cohn, T., Vylomova, K., Duh, K., Dyer, C. (2015). Depth-gated recurrent neural networks. arXiv preprint arXiv: arXiv:1508.03790v2

[42] Tjandra, A., Sakti, S., Manurung, R., Adriani, M., Nakamura, S. (2016). Gated recurrent neural tensor network. In 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, pp. 448-455. https://doi.org/10.1109/IJCNN.2016.7727233

[43] Schuster, M., Paliwal, K.K. (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11): 2673-2681. https://doi.org/10.1109/78.650093

[44] Alsarhan, T., Alawneh, L., Al-Zinati, M., Al-Ayyoub, M. (2019, October). Bidirectional gated recurrent units for human activity recognition using accelerometer data. In 2019 IEEE SENSORS, Montreal, QC, Canada, pp. 1-4. https://doi.org/10.1109/SENSORS43011.2019.8956560

[45] Li, P., Luo, A., Liu, J., Wang, Y., Zhu, J., Deng, Y., Zhang, J. (2020). Bidirectional gated recurrent unit neural network for Chinese address element segmentation. ISPRS International Journal of Geo-Information, 9(11): 635. https://doi.org/10.3390/ijgi9110635

[46] Zhang, D., Tian, L., Hong, M., Han, F., Ren, Y., Chen, Y. (2018). Combining convolution neural network and bidirectional gated recurrent unit for sentence semantic classification. IEEE Access, 6: 73750-73759. https://doi.org/10.1109/ACCESS.2018.2882878

[47] Liu, F., Zheng, J., Zheng, L., Chen, C. (2020). Combining attention-based bidirectional gated recurrent neural network and two-dimensional convolutional neural network for document-level sentiment classification. Neurocomputing, 371: 39-50. https://doi.org/10.1016/j.neucom.2019.09.012

[48] Wang, Z., Dixit, P., Chegdani, F., Takabi, B., Tai, B.L., El Mansori, M., Bukkapatnam, S. (2020). Bidirectional gated recurrent deep learning neural networks for smart acoustic emission sensing of natural fiber–reinforced polymer composite machining process. Smart and Sustainable Manufacturing Systems, 4(2): 179-198. https://doi.org/10.1520/SSMS20190042

[49] Namaa, D.S., AL-Zubaidi, M.M., AL-Rubai, H.K., Sabbah, M.A., Al-Janabi, T.Y., Hameed, S.N., Mahdi, A.A.A. (2019). Comparison between allele frequencies of several Strs Loci in Najaf City of Iraq and middle Province in Iraqi population. Indian Journal of Forensic Medicine & Toxicology, 13(4): 578. https://doi.org/10.5958/0973-9130.2019.00353.0