

FEDERATED IMPROVED RE-FED INCREMENTAL LEARNING WITH SPIKING LONG SHORT-TERM MEMORY NETWORK FOR SMART HEALTHCARE SYSTEMS

G. Sasikala¹ and G. Kalpana²

¹Department of Computer Science, Dr. G. R. Damodaran College of Science, India

²Department of Computer Science, Sri Ramakrishna College of Arts and Science for Women, India

Abstract

Worldwide Internet of Medical Things (IoMT) sector has been experiencing a vertiginous rate of evolution in the past few years, going from a little wristwatch to a large aeroplane. Smart Health Care (SHC) systems utilize innovation technologies like IoMT, cloud edge computing and Artificial Intelligence (AI). With connected wearable devices and quick replies, SHC improves healthcare management by making it more efficient, convenient, and personalized. Deep Learners (DL) in the cloud are trained using the data collected from these devices. These servers have a lot of memory and a lot of processing expenditures. By utilizing a decentralized architecture known as Federated Learning (FL), several edge clients can work together to build a unified DL model effectively protecting the privacy of their own data. When a model loses all memory of its prior training data after receiving fresh input is referred as Catastrophic Forgetting (CF) problem. When the data distribution on each device changes over time, this can happen in a FL environment. As a Federated Increment Learning (FIL) system, Re-Fed can reduce CF by letting all clients each client remembers past samples based on how important. However, discrepant arrival times of the new task and data from the malfunctioning clients are not handled by Re-Fed FIL. This paper propose a Federated Improved Re-Fed Incremental Learning (FIRFIL) which handle the above issue through temporally weighted aggregation. In this research, a Time-Invariant Stochastic Spiking Long Short Term Memory (TISSLSTM) is used in a FIRFIL scenario. Internet of Things (IoT) devices sent the data acquired from various wearable sensors including those for blood sugar, heart rate, and chest readings to edge devices equipped with TISSLSTM for training. In FIRFIL, every edge device uses its own private data set to train a local model. A centralized server receives the local models and merges them into one global model. Next, the edge devices are updated with trained global model once again. This loop is continued until either the global model converges, or specific amounts of training rounds have passed. Next, we use the trained model to forecast client-specific diseases based on incoming data. A temporal weighted aggregation model in the server handle temporally variants data from clients. The proposed model is simply known as FIRFIL-TISSLSTM. At last, the test result demonstrate that the proposed model achieves 95.09%, 95.25% and 94.28% of accuracy on Comprehensive Heart Disease Dataset, UCI Heart Disease Dataset and Kaggle Heart Disease Dataset respectively outperforming traditional models. Also, the proposed model records lower energy consumption values 89.7J, 80.3J, 86.1J of energy consumption and reduced latency values of 173.8ms, 162.5ms, 168.4ms of latency on same datasets highlight its efficiency compared to other standard models.

Keywords:

Internet of Medical Things (IoMT), Artificial Intelligence (AI), Smart Healthcare (SHC), Federated Incremental Learning (FIL), Time Invariant, Wearable Devices

1. INTRODUCTION

The ability of automated systems to improve accuracy, speed, and efficiency in human ailment classification is making them increasingly important in healthcare, leading to better patient outcomes [1]. The SHC system is generally linked to the IoMT to utilize and manage various smart gadgets. An intelligent prediction system based on the IoMT can be enhanced which it is integrated with machine and deep learning models [2] [3].

One of the primary duties in SHC is to keep an eye on patients' disease predictions. One important part of SHC is wearable technology. Electronic gadgets that patients wear and have various sensors allow for the monitoring, recording, and analysis of their health status. SHC actively manages and intelligently responds to the needs of the medical ecosystem by dynamically accessing information from wearable devices, connecting with healthcare managers/clinicians, materials, and institutions.

This class includes things like fitness bands, smartwatches, and devices that measure heart rate, blood pressure, glucose levels, and other vital signs [4]. The SHC system can simplify and secure the process of heart disease prediction using ML [5]. Health monitoring, habit tracking, and safety tests are all made easier with the help of SHC's helpful tools. Security and processing massive amounts of data are two major concerns, though.

This study focuses on the healthcare industry, where the difficulties are tackled in the FL setting [6]. FL is a machine learning method that let numerous users to work together to train a common model while maintaining the privacy and security of their own locally stored data. When it comes to medical records, this method offers a number of benefits. FL is a decentralized architecture that protects the privacy of local clients' data while enabling numerous edge clients to collaboratively learn a single DL model [7].

The majority of FL research has taken place in static settings, where the quantity of training data remains constant. Because of the problem known as catastrophic forgetting, which causes performance degradation on past jobs, FL has a hard time learning new data while preserving previous information in ML [8]. There were a lot of FIL approaches introduced to fix this. On the other hand, standard FIL approaches suffer from a terrible forgetfulness problem. To get over this problem, the FIL method known as Re-Fed [9] requires a high degree of correlation between cached samples and the statistical heterogeneity present in all client input. For Re-Fed to work, each client remembers past samples based on how important they are in their local dataset and how they are related to the global dataset. The client then uses the newly

acquired examples in addition to the cached samples to train the local model. However, discrepant arrival times of the new task and data from the malfunctioning clients are not handled by Re-Fed FIL.

Many DL models like RNN, LSTM, Bi-LSTM and GRU are used in FIL environment. These DL models very efficiently processes sequential data. However, computationally not efficient. By combining computationally efficient SNNs with LSTM and GRU models leverages the efficiency of one another. SNN-LSTM(S-LSTM) required more number of training parameters. In addition, SNN-GRU(S-GRU) spiking GRU model uses less training parameters but struggles to captures the long-term dependencies. An efficient hybrid DL model is required in FIL environment.

In this paper, FIRFIL – TISSLSTM model for efficient heart diseases prediction. This framework proposed FIRFIL to overcome the issues in Re-Fed FIL. In order to account personalized local factors like storage capacity, computational resources and CF problem, the newly arrived tasks with capturing more important cached samples in each client, the same functions of Re-Fed FIL are used. However, for handling varying arrival time of new tasks and task especially when received from faulty clients in server side, temporally weighted aggregation is proposed. The aggregation scheme synchronizes the data arrivals at various times from clients and handle inconsistencies. Because of the proposed aggregation, the FIL system becomes resistant to hostile clients and data heterogeneity. TISSLSTM is proposed and utilized in clients and server FIRFIL scenario. Because this TISSLSTM applies the non-linear spiking function individually at each time point, it reduces the training time and spike counts needed by the network during inference by removing it from the recurrence. In particular, for the prognosis of cardiovascular illnesses, this integration guarantees an accurate health care system with effective results.

2. LITERATURE REVIEW

Rehman et al [10] proposed a Federated learning Collaborative Clinical Cancer Diagnosis (FedCSCD) and Generative Adversarial Network (GAN) for clinical cancer diagnosis. In order to forecast the occurrence of cardiac problems, Bebertta et al. [11] presented a FEDEHR model for the heart diseases prediction that integrates FL with IoT-based electronic health records. Malwade et al. [12] combined FL and IoT-based HER data to create a shared learning method for predicting heart diseases. Manocha et al. [13] proposed an SVM classifier combined with Bi-directional LSTM encoder for ECG arrhythmia classification in a FL environment. Birari et al. [14] developed a Federated Transfer Learning (FTL) model with Adaptive Gradient Clipping (AGC) for heart disease prediction which ensures privacy in healthcare devices by integrating data from wearables, medical devices and EHRs. A healthcare system based on ECGs was created by Raza et al. [15] in a federated environment using Deep Convolutional Neural Networks (CNN) and Explainable AI (XAI) which classifies arrhythmias aiding clinical decision-making. Annappa et al. [16] developed Fedcure, an intelligent healthcare application framework for IoMT contexts that considers heterogeneity.

In order to improve the prediction performances in FL environment, hybrid DL models were utilized in literature. Khan et al. [17] created a FL model that integrated spiking neural network (SNN) with LSTM model for enabling fully supervised end-to-end learning for human activity detection. “Patel et al.” [18] A framework for Federated LSTM (F-LSTM) that safeguards user privacy by training various devices on distributed devices. A Spiking RNN(SRNN) with fully differentiable events was created by De et al. [19]. One of the problems with FL is catastrophic forgetting, which happens when models forget what they have learnt in the past while adjusting to new tasks. Other problems include communication overhead and data heterogeneity. FIL tackles these problems.

Dong et al. [20] created a Federated Class-Incremental Learning (FICL) model. In the FCIL setting, local clients continuously collect training data, while new clients with unseen classes can join at any time. Hu et al. [21] developed a FIL algorithm to randomly sampling samples from each client to maintain pre-training balance for preliminary period global model on the server. Psaltis et al. [22] presented a Federated knowledge distillation for representation based contrastive incremental learning (FedRCIL) which provides scalable approach for incremental learning in federated systems.

Li et al. [23] presented a Personalized Federated Domain-Incremental Learning (pFedDIL) model enables clients to choose an appropriate incremental task learning technique depending on their connection. Masum et al. [24] devised a Federated Few-Shot Class-Incremental Learning (FFSCIL) that handles CF issues with data privacy and data scarcity constraints. The aforementioned FIL environment, however, does not deal with data from clients that are malfunctioning or with new tasks that arrive at different times.

The fore-mentioned limitations of FIL models is resolved in this paper by developed improved Re-Fed model that intends to provide efficient heart prediction system.

3. PROPOSED METHODOLOGY

3.1 DATA PRE-PROCESSING

The collected dataset was having missing values, noise, distortions and class imbalance problems. K-Nearest Neighbors (KNN) imputation method used for filling missing data. Before applying in FIRFIL- TISSLSTM , data is normalized. When data is standardized, risk variables are scaled and numbers representing the difference among standard deviations around the mean are assigned. Eq.(1) gives the mathematical structure of standardization.

$$X = \frac{X - \text{Mean}(X)}{\text{Std}(X')} \quad (1)$$

where, X and X' is the original and standardized data respectively.

3.2 DATA CLASSIFICATION

The proposed FIRFIL- TISSLSTM for heart prediction systems is illustrated in below sections. Figure 1 displays the suggested FIRFIL-TISSLSTM for smart healthcare system heart disease prediction.

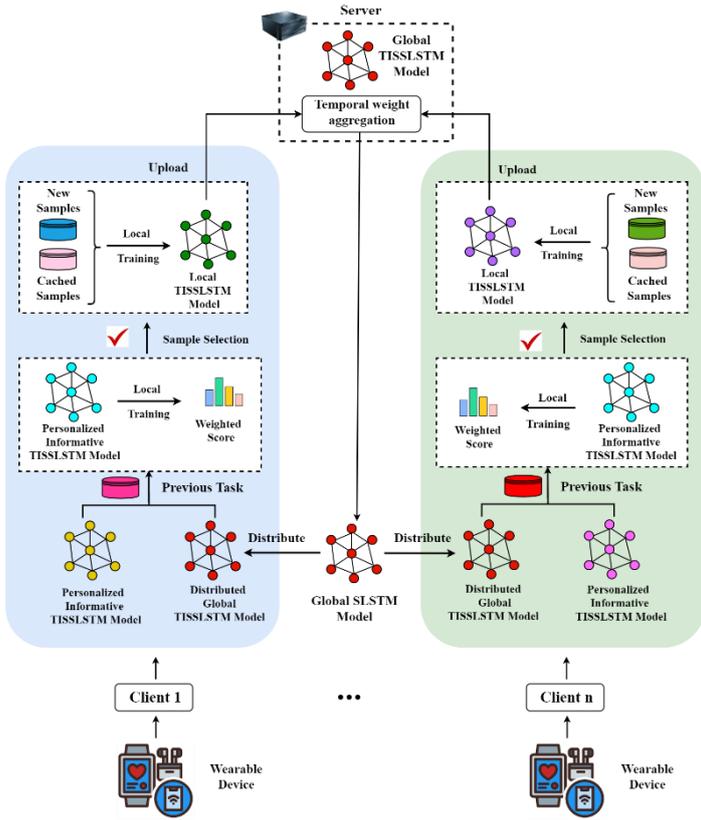


Fig.1. Proposed Architecture of FIRFIL – TISSLSTM model

3.3 DESIGN OF TISSLSTM

Synergistically combining the event-driven speed of time variant stochastic spiking model [25] with LSTM model. In the network, majority of neurons are Leaky-Integrate-And-Fire (LIF) neurons. As it processes incoming data and triggers an action potential when the membrane potential reaches a specific threshold, it explains how neurons behave.

The following equation provides a mathematical description of the LIF neuron model.

$$\tau \frac{dV(t)}{dt} = -[V(t) - V_{rest}] + RI(t) \quad (2)$$

In Eq.(2), $V(t)$ is the potential of neuron based on membrane at time t . $I(t)$ is the input current, R is the membrane resistance, τ is the time-dependent constant of the neuron's membrane, and V_{rest} is the membrane at its lowest potential. A neuron will fire, resetting $V(t)$, whenever the membrane potential $V(t)$ rises beyond a certain threshold. Neuronal integration and signal transmission in the brain can be simplified using this paradigm.

However, the LIF neuron cannot be time-parallelized since the non-linear firing function must be applied to the potential at time $t-1$ in order to calculate the membrane potential at time t . To overcome this limitation, we tweak the LIF neuron such that the recurrence does not include a non-linear spiking function. This is necessary because parallelization can only be achieved for linear systems.

The result is a variety of parts, including models for fixing LIF neurons, such as the Concurrent Leaky Integrator (CLI) neuron, and non-linear spiking function.

3.3.1 Concurrent Leaky Integrator:

The computation of membrane potential in a CLI is linearly time invariant since it does not depend on a non-linear spiking function. Therefore, representations that are not sequential are possible. Examine the LTI system delineated in Eq.(3), characterized by the state vector m , input vector v , and the state and input scalars a and b , respectively.

$$n[t] = a \cdot m[t-1] + b \cdot v[t] \quad (3)$$

This can also be written in Eq.(4) as,

$$m[t] = \sum_{j=1}^t a^{t-j} \cdot v[j] \quad (4)$$

Using $h = [a^{t-1}b, a^{t-2}b, \dots, a^0b]$, which can be expressed in Eq.(4) as a non-sequential vector multiplication.

$$m[t] = h \cdot v \quad (5)$$

Consequently, Eq.(5) determines the state of m at time n . A convolution of values among h and v can be performed for each state.

$$m[1:t] = h * v \quad (6)$$

Employing the convolution theorem, Eq.(6) can be efficiently evaluated in the Fourier domain, whereby \mathcal{F} denotes the Fourier transform.

$$m[1:t] = \mathcal{F}^{-1} \{ \mathcal{F} \{h\} \cdot \mathcal{F} \{v\} \} \quad (7)$$

The previous equations demonstrated how to parallelize a sequential LTI system Eq.(8). This approach can consequently be used to the LI neuron Eq.(7).

$$i[1:t] = \mathcal{F}^{-1} \{ \mathcal{F} \{ \ell \} \cdot \mathcal{F} \{ x \} \} \quad (8)$$

$$u[1:t] = \mathcal{F}^{-1} \{ \mathcal{F} \{ y \} \cdot \mathcal{F} \{ i[1:t] \} \} \quad (9)$$

where,

$$\ell = [\alpha^{t-1}, \alpha^{t-2}, \dots, \alpha^0],$$

$$y = [\beta^{t-1}(1-\beta), \beta^{t-2}(1-\beta), \dots, \beta^0(1-\beta)],$$

i is the input current and x is the membrane time constant. Finally, Eq.(8) and Eq.(9) can be combined as follows,

$$u[1:t] = \mathcal{F}^{-1} \{ \mathcal{F} \{ k \} \cdot \mathcal{F} \{ \ell \} \cdot \mathcal{F} \{ x \} \} \quad (10)$$

It is possible to quickly and parallelly determine the LI membrane potential using the Eq.(10). The next step in implementing the suggested SPSN is to show how the spikes are generated.

3.3.2 Stochastic Firing:

An approximation of the stochastic spike production observed experimentally in biological neurons is achieved by modelling the firing process with a firing probability formula based on membrane potential. Eq.(11) describes this function using a , b , and c as matching parameters.

$$\rho(t) = \frac{1}{a} \exp\left(\frac{u(t)-b}{c}\right) \quad (11)$$

A stochastic firing criterion, $\rho(t) = f(u(t) - u_{th})$ where f is the escape function, is used to generate spikes in this model instead of a tight firing threshold. Two stochastic spiking

functions, each applicable at distinct times, are shown in this study.

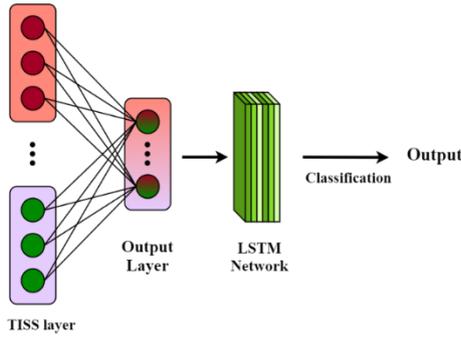


Fig.2. Proposed TISSLSTM model

One kind of RNN that can learn long-term dependencies is the LSTM network [26], which is used for sequence prediction tasks using TIS input. LSTMs address the problems of standard RNNs, such as disappearing and exploding gradients, by deploying gates that govern the flow of information. The three types of gates that comprise each LSTM unit (cell) are input, output, and forget gates. These gates regulate the flow of data into and out of the LSTM, which allows it to store data for extended periods of time.

3.4 FEDERATED INCREMENTAL LEARNING (FIL)

FIL cases in which edge clients might not have sufficient storage to keep complete data. The FIL framework ensures clients save crucial samples for later playback by caching a subset of previously collected samples based on their global and local relevance. Clients then use these newly acquired examples in addition to the cached samples to train the local model. Within the conventional IL, namely in a non-federated setting, a model is trained via a series of streaming tasks $\{T^1, T^2, \dots, T^m\}$, where T^t signifies the t^{th} task in the dataset. In this case, $T^t = \left\{ \left(x_i^{(t)}, y_i^{(t)} \right) \right\}_{i=1}^{N^t}$, where $x_i^{(t)} \in X^t$ and $y_i^{(t)} \in Y^t$ are N^t pairs of sample data. The domain set and label space for the t^{th} task are denoted by X^t and Y^t respectively. The current task contains $|Y^t|$ classes and $Y = \bigcup_{t=1}^n Y^t$, where Y represents the total classes of all time. In a similar vein, the complete domain space for activities across time can be represented by $X = \bigcup_{t=1}^n X^t$. Pay attention to two kinds of IL situations: In a class-incremental job, all tasks are performed on the same domain space, where $X = \bigcup_{t=1}^n X^t$ for every t in the interval $[n]$. As the learning challenges are introduced, the number of classes can change, meaning $Y^1 \neq Y^t$, where t is an integer from 1 to n .

Tasks in the domain-incremental model have an equal number of classes, $Y^1 = Y^t$ where t is an integer from 1 to n . The client must acquire new skills when the tasks are introduced, all the while their domain and data distribution are altered, so $X^1 \neq X^t$, for every $t \in [n]$. Think about IL more in a federated context. Client k only has access to the local confidential streaming tasks $\{T_k^1, T_k^2, \dots, T_k^n\}$, hence in order to train a global model for K total

clients, we must accept this assumption. The objective is to train a global model w^t across all t tasks when the t^{th} task arrives, even if clients can remember all samples from prior tasks in their cache.

They may express it as Eq.(12), which is $T^t = \left\{ \sum_{n=1}^t \sum_{k=1}^K T_k^n \right\}$.

$$w^t = \arg \min_w \sum_{n=1}^t \sum_{k=1}^K \sum_{i=1}^{N_k^n} \frac{1}{T^t} \cdot \ell \left(f_{w_k} \left(x_{k,n}^{(i)} \right), y_{k,n}^{(i)} \right) \quad (12)$$

The cross-entropy loss is denoted by $l(\cdot)$ and the result of the model w_k in client k is represented by f_{w_k} . Then, because shared edge devices have limited storage capacity, clients store incomplete samples in their caches for later use. Assume that each client has a storage capacity of M samples and must cache $M - N_k^t$ samples from previous jobs at time $(t-1)$. This is

represented as $T_{k,\text{cached}}^{(t-1)} = \sum_{i=1}^{M-N_k^t} \left(\bar{x}_{k,t-1}^{(i)}, \bar{y}_{k,t-1}^{(i)} \right)$. Using both preloaded samples and the t^{th} new task, the objective is to train a global model w^t , which may be expressed as Eq.(13).

$$w^t = \arg \min_w \sum_{k=1}^K \sum_{i=1}^M \frac{1}{T_{k,\text{local}}^t} \cdot \ell \left(f_{w_k} \left(\tilde{x}_{k,t}^{(i)} \right), \tilde{y}_{k,t}^{(i)} \right) \quad (13)$$

where, $T_{k,\text{local}}^t = T_{k,\text{cached}}^{(t-1)} + T_k^t = \sum_{i=1}^M \left(\tilde{x}_{k,t}^{(i)}, \tilde{y}_{k,t}^{(i)} \right)$

3.5 RE-FED: FRAMEWORK FOR FIL

In order to facilitate the arrival of new tasks, Re-Fed prioritizes samples based on their importance and coordinates with clients to cache crucial earlier samples using limited local storage. During communication rounds, clients train local models using private tasks, while the server compiles all models. Each client uses both global and local models to train an additional informative model using past local samples, ensuring efficient communication [9].

The sample relevance scores are determined by keeping track of the gradients averages of each sample when this sort of model is changed. As a general rule, in a FL scenario, every client has access to both their local model, which is location-specific, and their global model, which is FL-wide. This means that clients can cache samples that have greater relevance ratings before going on to the next assignment.

Using both local and global models, one may easily determine the relative relevance of a sample and then use these scores to inform data response from cache samples. Afterwards, one can expand upon this concept by incorporating the following features:

The global model's gradient norm can be computed locally without training, and local models from clients aggregate it. The relative weight of local and global data in the sample can be adjusted using a control mechanism. In order to achieve the aforementioned goals, a Personalized Informative Model (PIM) is presented for every customer. This model takes in data from both the local and global models. After that, it is suggested a ratio factor to modify the local-to-global information proportion. Imagine for a moment that client k has received the global model w^{t-1} and the t^{th} new task has arrived. In s iterations, the clients update PIM v_k^{t-1} using prior local samples $T_{k,\text{local}}^{t-1}$ according to Eq.(14).

$$v_{k,s}^{(t-1)} = v_{k,s-1}^{(t-1)} - \eta \left(\sum_{i=1}^M \nabla \ell \left(f_{v_{k,s-1}^{(t-1)}} \left(\tilde{x}_{k,t-1}^i \right), \tilde{y}_{k,t-1}^i \right) + q(\lambda) \left(v_{k,s-1}^{(t-1)} - w^{(t-1)} \right) \right) \quad (14)$$

The rate to control the step size of the update, η , and the function $q(\lambda) = 1 - \lambda - 2\lambda$ for every $\lambda \in (0, 1)$. The hyper-parameter λ modifies the equilibrium of the update by adjusting the proportion of local and global information. Likewise, the momentum component is represented by $(\lambda)(v_{k,s-1}^{(t-1)} - w^{(t-1)})$. In order to influence the update of PIM $v_k^{(t-1)}$, it uses data from the global model w^{t-1} . The hyper-parameter λ , which is between 0 and 1, determines the weight of this momentum component. Recovering the global model w^{t-1} is PIM's main aim when λ is around 0. Basically, it will adjust its behavior to match the facts from across the world. Conversely, a greater focus on local training is caused by an increase in λ .

An upper bound is defined by the sample gradient norm as given in Eq.(11), and it is directly related to the difference between the gradient of the loss function regardless of a sample $(\tilde{x}_{k,t-1}^{(i)}, \tilde{y}_{k,t-1}^{(i)})$. The training dynamics are best preserved when samples are cached according to sample gradient norms, as this method has the least impact on the gradient. PIM uses both local and global models, so a sample with a higher gradient norm is more likely to fit the job using both sets of information. This effect may be more pronounced during early training iterations. Therefore, to determine the sample importance, the gradient norm are integrated while training PIM, with a focus on the early stages of training, and use Eq.(15)

$$I(\tilde{x}_{k,t-1}^{(i)}) = \sum_{p=1}^s \frac{1}{p} \cdot G^p(\tilde{x}_{k,t-1}^{(i)}) \quad (15)$$

In iteration $p \in [1, s]$, each client continues to train the local model w_k^t using local samples $T_{k,\text{local}}^t$ in accordance with Eq.(13), following the caching of significant samples that have greater significance scores.

$$w_{k,p}^t = w_{k,p-1}^t - \eta \sum_{i=1}^M \nabla \ell \left(f_{v_{k,p-1}^t} \left(\tilde{x}_{k,p-1}^{(i)} \right), \tilde{y}_{k,t}^{(i)} \right) \quad (16)$$

The TISSLSTM model is a strong combination of SNN's event-driven performance and LSTM's sequential modelling capacity.

D. Temporally Weighted Aggregation

In order to prioritize new arrivals, temporal weight aggregation model is integrated is applied in Re-Fed to enhance the synchronization and address the issues in faulty client and task arrival timings. The temporally weighted aggregation of the local and global models on the servers is given in Eq.(17)

$$w_{k,p}^t = \frac{\square \alpha_t - \alpha_t^{(k,p)} \square^{-1}}{\sum_{y=1}^Y \square \alpha_t - \alpha_t^{(k,p)} \square^{-1}} \quad (17)$$

where, α_t represents the adaptive weighting coefficient, $\alpha_t^{(k,p)}$ represents the present model parameter (p) of k^{th} client. Assuming the time effect during the federated communications, the temporally weighted aggregation of the local and global models on the server is termed as Eq.(18):

$$\alpha_{t+1} = \sum_{y=1}^Y \frac{n_k}{n} \left(\frac{e}{2} \right)^{-(t-t^{(k,p)})} \cdot \alpha_t^{(k,p)} \quad (18)$$

where, t is the present update round, $t^{(y)}$ will be the update round of the newest $\alpha^{(y)}$ and e is the constant. $\alpha_{(t+1)}$ is the upgrade global model parameter after aggregation. In addition to the temporal factors, the accuracy of the local and global models can also act as the key reference of adjusting the weights adaptively. The training process of FIRFIL- SLSTM is given in below a FIRFIL- SLSTM algorithm.

Algorithm 1 FIRFIL – TISSLSTM training

1. Data: Client $i = 1, 2, \dots, N$, initial variables for the model w_0 ,
2. Results: Parameters of the trained model w
3. Method: Initialisation
4. For every client i work with simultaneously,
5. $D(i) \leftarrow$ local dataset
6. $w^{(i)} \leftarrow w_0$ {Get the local model started}
7. end for
8. Method: FL training
9. for iterations $r = 1, 2, \dots, R$, execute
10. A subset of devices S_t is randomly selected by the server.
11. Call ServerUpdate
12. send w_{t-1} to all clients
13. end for
14. For every chosen client k from the set S_t simultaneously
15. In each cycle, update v_k^{t-1} locally.
16. As v_k^{t-1} is being updated, do
17. Determine the sample's significance score following a total of s iterations $(\tilde{x}_{k,t-1}^{(i)}, \tilde{y}_{k,t-1}^{(i)})$
18. end for
19. Keep earlier samples with higher priority scores in a cache; //Usi> n^g previously stored samples and a fresh task to train a local model
20. LocalTraining(v, D)
21. Reply to the server with the model w_k^t
22. end for
23. end
24. Method: LocalTraining(w, D)
25. Set up the learning rate η and local parameters $w^{(i)}$, to their initial values.
26. For iteration at time interval $t = 1, \dots, T$ do
27. Find the local gradient
28. Calculate Δw from $w^{(i)}(t)$
29. end for
30. Return Δw
31. Method: ServerUpdate($\{\Delta w(i)\}_{i=1}^N$)
32. $w \leftarrow$ aggregate($\{\Delta w(i)\}$) \\ temporal weight aggregation
33. return updated model parameters w
34. Rank the new arrivals

4. RESULTS AND DISCUSSION

4.1 DATA AGGREGATION

Comprehensive Heart Disease Dataset [27]: Eleven criteria from 1190 cases make up this dataset, which is a combination of five popular heart disease datasets.

1. **UCI Heart Disease Dataset [28]:** This dataset contains a variety of numerical variables utilized in multivariate numerical data analysis. It includes 14 variables shown in table1. As shown in Table.1, these characteristics allow for the analysis of data patterns and correlations, and they also provide information on several facets of cardiovascular health.
2. **Kaggle Dataset [29]:** The 76 qualities include the anticipated attribute, which is a subset of 14 of those attributes. The presence of cardiac disease in the patient is what the "target" field is referring to. Its value is an integer. An illness is present if the value is 1 and absent if it is 0.

4.2 EXPERIMENTAL SETUP

Simulation codes for proposed and current protocols were run on a laptop with the following specifications: 1TB HDD, 4GB RAM, Intel® Core™ i5-4210 CPU @ 2.80GHz, and Windows 10 64-bit. Python 3.7 was used to run the simulation. This framework encompasses 3 dataset illustrated in section 4.1. The proposed model adjust the hyperparameter $\lambda=0.5$ to facilitate identically distributed data across 20 clients. Also, the exemplary memory size (M) is set as $\{500,1000,1500,2000\}$, m is set to be the twice of the total number of classes from dataset in every task. The Table.2 depicts the parameter configuration of the proposed model.

Table.1. Dataset Description (UCI Heart Disease Dataset)

| Variable Name | Type |
|-----------------------------------------------------------------------------|-------------|
| Existence or Nonexistence of Cardiac Disease (Target variable) | Integer |
| Age | Integer |
| Chest Pain Type | Categorical |
| Serum Cholesterol | Integer |
| Angina Induced by Exercise | Categorical |
| Fasting Glucose Level | Categorical |
| Peak Heart Rate Attained | Integer |
| Number of Major Vessels | Integer |
| Previous peak (ST segment depression elicited by activity compared to rest) | Integer |
| Resting Blood Pressure | Integer |
| Resting Electrocardiographic Results | Categorical |
| Gender | Categorical |
| Slope of the Peak Exercise ST Segment | Categorical |
| Thalassemia | Categorical |

Table.2. Parameter Configuration (TISSLSTM)

| Parameters | Range |
|----------------------|-------|
| No. of hidden layers | 3 |
| Training rate | 0.001 |
| Dropout rate | 0.5 |
| Momentum | 0.7 |
| Number of epochs | 120 |
| Batch size | 64 |
| Optimizer | Adam |

4.3 PERFORMANCE ANALYSIS

In this section, the proposed FIRFIL – TISSLSTM model is evaluated on three dataset evaluated on different metrics. The proposed TISSLSTM is compared with standard models like LSTM, Bi-LSTM, Spiking LSTM (S-LSTM), Invariant LSTM (I-LSTM) [30] and TISSLSTM. The metrics applied for performance analysis are listed below.

The appropriate predictions made by proposed classifier is evaluated by Accuracy (Acc) which is expressed as Eq.(19)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (19)$$

In Eq.(14), True Positive (TP) and True Negative (TN) signifies the model's accurate prediction of the negative case, while the positive case is predicted accurately as well. False Positive (FP) and False Negative (FN) denote instances where the model erroneously predicts a positive case and a negative case, respectively.

The proportion of TP within a set of projected positives is called precision (Pre). The mathematical expression for in Eq.(20).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (20)$$

One way to measure the impact of the overall number of FN occurrences on the total number of TP instances is by using Recall (Rec) or Sensitivity (Sen). It can be expressed mathematically as Eq.(21).

$$\text{Recall} = \frac{TP}{TP + FN} \quad (21)$$

F1-score (F1): It is calculated by Eq.(22)

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (22)$$

In case of FIL, the proposed FIRFIL model is compared with FedRCIL [18], FedDIL [19], FFSCIL [20] Re-Fed [26] using different metrics give below.

Energy consumption (Joule (J)): It is referred to as the average energy disbursed by all sensor nodes per round.

$$E_{\text{mean}} = \frac{1}{N} \sum_{x=1}^N (E_{\text{init}}^x - E_{\text{res}}^x) \quad (23)$$

In Eq.(23), E_{init}^x is the original energy of x , E_{res}^x is the residual energy of x after simulation runs, and N represents the overall quantity of sensor nodes.

Using comparable performance indicators, the suggested model's effectiveness is evaluated by contrasting it with the current arrhythmia classification model. 30% of the gathered dataset is used for testing, while 70% is used for training. Additionally, using the same assessment criteria, a comparison study is conducted to determine how much the TISSLSTM model has improved over the current models, such as LSTM, Bi-LSTM, SLSTM and I-LSTM.

The Table.3 presents a comparison of different heart disease prediction models on Wearable device data. The proposed model archives highest accuracy, precision, recall and f1-score values than LSTM, Bi-LSTM, S-LSTM and I-LSTM models respectively on comprehensive, UCI and kaggle heart disease datasets. The proposed FRFIL architecture enhances the model's adaptability from multiple angles. These advantages markedly improve the effectiveness and dependability of the model in identifying and categorizing arrhythmias from wearable sensor data. The Table.4 depicts the comparison of different network metrics like energy consumption and latency of proposed and existing models on three datasets. In this analysis, the proposed model shows lower results on comprehensive, UCI and kaggle heart disease datasets.

Table.3. Performance analysis of proposed and standard models on three dataset

| Dataset | Metrics | Models | | | | |
|----------------------------------------------------|---------|--------|---------|--------|--------------|--------------|
| | | LSTM | Bi-LSTM | S-LSTM | I-LSTM | TISSLSTM |
| Compre- -hensive Heart Disease Dataset | Acc | 82.75 | 85.49 | 88.67 | 94.28 | 96.12 |
| | Pre | 81.33 | 83.01 | 86.52 | 92.19 | 94.88 |
| | Rec | 80.6 | 83.95 | 87.18 | 93.43 | 95.57 |
| | F1 | 81.81 | 84.69 | 87.97 | 93.24 | 95.09 |
| UCI Heart Disease Dataset | | LSTM | Bi-LSTM | S-LSTM | I-LSTM | TISSLSTM |
| | Acc | 80.12 | 84.05 | 88.37 | 91.48 | 95.33 |
| | Pre | 82.62 | 86.64 | 90.13 | 94.24 | 96.24 |
| | Rec | 81.07 | 87.59 | 89.47 | 92.06 | 94.74 |
| F1 | 83.27 | 85.25 | 89.67 | 92.06 | 95.24 | |
| Kaggle Heart Disease Dataset | | LSTM | Bi-LSTM | S-LSTM | I-LSTM | TISSLSTM |
| | Acc | 81.67 | 86.43 | 89.65 | 91.43 | 94.28 |
| | Pre | 80.23 | 87.56 | 91.28 | 94.29 | 95.65 |
| | Rec | 82.52 | 85.12 | 89.17 | 90.14 | 96.14 |
| F1 | 83.07 | 86.32 | 89.49 | 92.75 | 94.28 | |

Table.4. Analysis of Energy Consumption and Latency for proposed and standard models

| Dataset | Metrics | Models | | | | |
|--------------------------------|------------------------------|---------|--------|--------|--------|--------------|
| | | FedRCIL | FedDIL | FFSCIL | Re-Fed | FIRFIL |
| Comprehensive Heart Disease | Energy Consumption (J) | 120.4 | 114.8 | 110.3 | 102.6 | 89.7 |
| UCI Heart | | 105.2 | 99.6 | 95.1 | 91.4 | 80.3 |
| Kaggle Heart | | 112.7 | 108.2 | 104.5 | 98.3 | 86.1 |
| Comprehensive Heart Disease | Latency (ms) | 235.6 | 222.1 | 210.4 | 198.2 | 173.8 |
| UCI Heart | | 218.3 | 205.7 | 198.9 | 187.6 | 162.5 |
| Kaggle Heart | | 225.0 | 212.6 | 202.3 | 191.9 | 168.4 |

Table.5. Computational Metrics Evaluation of 20 clients (Exemplary memory size M, incremental task T= 20) *(Average Test Accuracy = ATA (all dataset); Training Time (TT); Inference Time (IT); Seconds (s))

| Model | ATA (%) | TT (s) | IT (s) | ATA (%) | TT (s) | IT (s) | ATA (%) | TT (s) | IT (s) | ATA (%) | TT (s) | IT (s) |
|---------------|-------------|--------------|--------------|-------------|--------------|--------------|-------------|--------------|--------------|--------------|--------------|--------------|
| | M = 500 | | | M = 1000 | | | M = 1500 | | | M = 2000 | | |
| FedRCIL | 32.5 | 315.7 | 115.3 | 29.8 | 621.2 | 233.3 | 25.4 | 933.1 | 411.5 | 21.4 | 1245.9 | 778.9 |
| FedDIL | 33.9 | 298.2 | 127.6 | 32.2 | 590.5 | 220.1 | 28.1 | 880.4 | 397.6 | 23.50 | 1190.3 | 674.1 |
| FFSCIL | 36.5 | 275.8 | 143.9 | 35.0 | 555.3 | 208.4 | 30.6 | 833.6 | 366.7 | 26.8 | 1132.8 | 668.9 |
| Re-Fed | 39.2 | 248.6 | 129.8 | 38.9 | 497.5 | 187.6 | 33.3 | 750.2 | 315.3 | 28.7 | 1016.9 | 602.4 |
| FIRFIL | 41.7 | 210.4 | 133.4 | 40.4 | 425.9 | 160.8 | 35.6 | 640.1 | 308.6 | 30.22 | 875.2 | 552.6 |

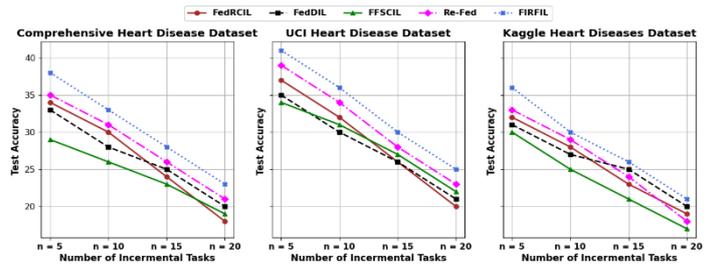


Fig.3. Performance for number of incremental tasks n for three datasets.

4.4 COMPUTATIONAL ANALYSIS

In this section, the proposed and existing models are performed under varying exemplary memory size ((M = 500, 1000, 1500, 2000) configurations with incremental task (T = 20), as illustrated in Table.5. FIRFIL consistently achieves the highest ATA at all memory sizes (41.7% at M = 500, 40.4% at M = 1000, 35.6% at M = 1500, and 30.22% at M = 2000), showcasing its superior performance. Although its TT and IT are higher than other models, which are not extensive, maintaining a balance between accuracy and efficiency. In comparison, FedRCIL exhibits the lowest ATA at all memory sizes (32.5% at M = 500, 29.8% at M = 1000, 25.4% at M = 1500, and 21.4% at M = 2000), alongside significantly higher computational times, especially with larger memory sizes. FedDIL and FFSCIL perform moderately in terms of ATA but still fall short of FIRFIL's consistency and efficiency. Re-Fed also shows good results, but its ATA decreases more noticeably as memory size increases, further solidifying FIRFIL as the top contender for optimal performance in the context of these models. The proposed model highlights FIRFIL's suitability for resource-constrained applications, especially in real-time smart healthcare systems, due to its efficient computational and memory capabilities.

The Fig.3 depicts the performance analysis for number of incremental task (n = 20) of proposed and existing models on three dataset. As the number of incremental tasks increases from 5 to 20, all methods show a decline in test accuracy, highlighting the challenge of maintaining performance in incremental learning task. Among the methods, FIRFIL consistently outperforms others across all datasets, demonstrating its robustness and effectiveness. Re-Fed and FFSCIL also perform relatively well but fall short of FIRFIL's accuracy. FedRCIL and FedDIL show the lowest accuracy, indicating limitations in adapting to incremental learning tasks.

Table.6. t-test Result of Proposed and Existing models

| Metric | Model | t | Diff (df) | Sig. (2-tailed) | Mean | Mean Diff | 95% CI [Lower, Upper] |
|--------------------|---------------|---------------|-------------|-----------------|--------------|--------------|-----------------------|
| Test Accuracy (%) | FedRCIL | 23.078 | 2000 | 0.0000 | 27.27 | 21.70 | [28.88, 29.52] |
| | FedDIL | 35.932 | 2000 | 0.0000 | 29.30 | 17.67 | [25.26, 28.09] |
| | FFSCIL | 33.945 | 2000 | 0.0000 | 30.97 | 15.00 | [23.02, 26.98] |
| | Re-Fed | 27.415 | 2000 | 0.0000 | 35.02 | 13.95 | [25.43, 27.47] |
| | FIRFIL | 32.675 | 2000 | 0.0000 | 34.97 | 9.96 | [24.85, 26.32] |
| Training Time (s) | FedRCIL | 74.174 | 2000 | 0.0000 | 78.97 | 42.07 | [79.14, 83.55] |
| | FedDIL | 69.130 | 2000 | 0.0000 | 79.85 | 34.95 | [74.28, 78.87] |
| | FFSCIL | 72.904 | 2000 | 0.0000 | 69.37 | 30.47 | [70.80, 73.10] |
| | Re-Fed | 64.010 | 2000 | 0.0000 | 62.30 | 24.40 | [69.53, 72.42] |
| | FIRFIL | 75.258 | 2000 | 0.0000 | 73.91 | 20.68 | [62.58, 65.22] |
| Inference Time (s) | FedRCIL | 51.899 | 2000 | 0.0000 | 54.75 | 25.90 | [49.28, 52.33] |
| | FedDIL | 47.436 | 2000 | 0.0000 | 56.85 | 21.32 | [44.87, 47.07] |
| | FFSCIL | 52.652 | 2000 | 0.0000 | 55.97 | 17.12 | [40.11, 43.89] |
| | Re-Fed | 48.690 | 2000 | 0.0000 | 53.77 | 14.73 | [39.17,42.08] |
| | FIRFIL | 55.137 | 2000 | 0.0000 | 52.85 | 10.99 | [37.03, 40.18] |

4.5 STATISTICAL ANALYSIS

For the statistical analysis, the proposed and the standard model are evaluated in terms of Confidence Intervals (CIs) and T-test. The CIs relies on the range within the true value (accuracy), which is likely expected to fall within a certain level of confidence (usually 95%). It is expressed in Eq.(24)

$$CI = \bar{x} \pm t_{\alpha/2, n-1} \cdot \frac{s}{\sqrt{n}} \tag{24}$$

where, x is the mean sample, s and n represents the sample standard deviation and sample size. $t_{\alpha/2, n-1}$ will be the t-score based on the confidence level.

A t-test is a statistical procedure employed to ascertain whether a significant difference is present in the means of two groups. It evaluates whether the differences noticed are attributable to chance or signify a genuine effect. It is presented in Eq.(24)

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{25}$$

where, x_1, x_2 represents the samples, s_1, s_2 are the standard deviations, x_1, x_2 defines the number of samples in a group.

The Table.6 shows t-test findings comparing the performance of proposed and existing models across three crucial metrics: training time, inference time and memory size. For test accuracy, FIRFIL and Re-Fed achieve superior performance, while FedRCIL ranks lowest result. Despite FIRFIL having slightly lower test accuracy than Re-Fed, its mean difference and confidence interval suggest a more consistent performance. In terms of training time, FIRFIL is efficient, showing the second-lowest average after Re-Fed making it suitable for time-sensitive applications. FedRCIL and FedDIL require the most training time, indicating heavier computational demands. For inference time, FIRFIL again leads with the shortest average, followed

closely by Re-Fed, while FedRCIL and FedDIL lag behind. Overall, this demonstrates that FIRFIL is not only quicker, but highly suitable to apply in resource-constrained in real-time healthcare applications.

4.6 DISCUSSION

The ablation study in previous sections methodically assesses the importance of each element in the FIRFIL–TISSLSTM model. The substitution of TISSLSTM with traditional LSTM-based models resulted in a decline in overall classification accuracy, underscoring TISSLSTM’s superiority in effectively learning temporal relationships. Removing the FIRFIL component resulted in increased latency and energy consumption due to inefficient synchronization and sample caching. Excluding the temporal weighted aggregation caused model performance degradation under asynchronous client updates. Reducing the memory size (M) significantly impacted accuracy, indicating its role in incremental learning. Replacing stochastic spiking neurons with deterministic units diminished generalization in dynamic conditions. Without FIL, overfitting occurred due to reliance on localized data. The complete FIRFIL–TISSLSTM model achieved the highest performance across all metrics. These results demonstrate that each module plays a critical role in enhancing accuracy, efficiency, and robustness for smart healthcare applications.

5. CONCLUSION

This paper presents the FIRFIL - TISSLSTM model for heart disease prediction. TISSLSTM is applied in a FIRFIL situation. IoT devices sent data collected from different wearable sensors, including blood sugar, heart rate, and chest measurements, to edge devices equipped with TISSLSTM for training. In FIRFIL, each edge device trains a local model using its own private dataset. A centralized server accepts local models and integrates them into a single global model. Next, the edge devices are updated with the trained global model once again. This loop is repeated until either the global model converges or a set number of training rounds have elapsed. Next, we utilize the trained model to predict client-specific illnesses based on incoming data. A temporal weighted aggregation model on the server handles temporally varying data from clients. Finally, the experimental result devises that the proposed model attains accuracy of 95.09%, 95.25%, and 94.28% on three datasets respectively which outperforms other models. In addition, the proposed model records lower energy consumption values of 89.7J, 80.3J and 86.1J, along with reduced latency of 173.8 ms, 162.5 ms, and 168.4 ms on the same datasets, underscoring its efficiency compared to other standard models. This demonstrates the models ability to deliver high performance heart diseases prediction system within resource-constrained health care environments.

REFERENCES

[1] M.A. Karp, “Figures of Contrast in Animal Farm by George Orwell: Lexical and Semantic Features of Oxymoron, Paradox and Antithesis”, Baltija Publishing, pp. 232-249, 2021.

- [2] M. La Pietra and F. Masini, "Oxymorons: A Preliminary Corpus Investigation", *Proceedings of Workshop on Figurative Language Processing*, pp. 176-185, 2020.
- [3] S.N. Safarovna, "Oxymoron as a Method of Poetic Perception of the World", *Journal of Critical Reviews*, Vol. 7, No. 7, pp. 340-344, 2020.
- [4] Y.C. Tsao, S.L. Chu, H.C. Liao and S.Y. Huang, "Application of Innovative Oxymorons in Product Design", *International Journal of Organizational Innovation*, Vol. 13, No. 1, pp. 340-355, 2020.
- [5] B.R. Bageshwar, "A Study of Literary Devices used in the Posts on Instagram", *International Journal of English Literature and Social Sciences*, Vol. 6, No. 1, pp. 414-421, 2021.
- [6] K.V. Fazlitdinovna, "Oxymoron and its Methodological-Semantic Features", *Bulletin of Science and Education*, Vol. 7, No. 2, pp. 114-119, 2020.
- [7] L. Sakaeva and L. Kornilova, "Structural Analysis of the Oxymoron in the Sonnets of William Shakespeare", *Journal of History Culture and Art Research*, Vol. 6, No. 5, pp. 409-414, 2017.
- [8] M.E. Alwana, "A Semantic and Pragmatic Study of the Oxymoron in Selected Poems", *International Journal of Innovation, Creativity and Change*, Vol. 13, No. 7, pp. 290-300, 2020.
- [9] W.I. Cho, W.H. Kang, H.S. Lee and N.S. Kim, "Detecting Oxymoron in a Single Statement", *Proceedings of International Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment*, pp. 1-5, 2017.
- [10] J.H. Ruiz, "Paradox and Oxymoron Revisited", *Procedia-Social and Behavioral Sciences*, Vol. 173, pp. 199-206, 2015.
- [11] C.B. De Haan, G. Chabre, F. Lapique, G. Regev and A. Wegmann, "Oxymoron, a Non-Distance Knowledge Sharing Tool for Social Science Students and Researchers", *Proceedings of International Conference on Supporting Group Work*, pp. 219-228, 1999.
- [12] R.W. Gibbs and L.R. Kearney, "When Parting is Such Sweet Sorrow: The Comprehension and Appreciation of Oxymora", *Journal of Psycholinguistic Research*, Vol. 23, No. 1, pp. 75-89, 1994.
- [13] Y. Shen, "On the Structure and Understanding of Poetic Oxymoron", *Poetics Today*, Vol. 8, No. 1, pp. 105-122, 1987.
- [14] L.S.E. Muhammed and L.J.A. Meftin, "A Study of Pleonasm with Reference to Some Biblical Verses", *Journal of Al-Qadisiya University*, Vol. 19, No. 1, pp. 7-24, 2016.
- [15] M. Stamborg and P. Nugues, "Statistical Identification of Pleonastic Pronouns", *Proceedings of International Conference on Swedish Language Technology*, pp. 67-68, 2012.
- [16] S. Haider, M. Tanvir Afzal, M. Asif, H. Maurer, A. Ahmad and A. Abuarqoub, "Impact Analysis of Adverbs for Sentiment Classification on Twitter Product Reviews", *Concurrency and Computation: Practice and Experience*, Vol. 33, No. 4, pp. 1-15, 2021.
- [17] A. Aytan, B. Aynur, P. Hilal, E. Aytac and A. Malahat, "Euphemisms and Cho Dysphemisms as Language Means Implementing Rhetorical Strategies in Political Discourse", *Journal of Language and Linguistic Studies*, Vol. 17, No. 2, pp. 741-754, 2021.
- [18] E.D. Oktaviyani, "Redundancy Detection of Sentence Pairs in the Software Requirements Specification Documents with Semantic Approach", *Proceedings of International Conference on Informatics, Engineering, Science and Technology*, Vol. 879, No. 1, pp. 1-8, 2020.
- [19] D. Sethi, "Workplace Impression Management through Pleonastic English Words/Phrases in E-Mail Communication? Evidence from India", *International Journal of Indian Culture and Business Management*, Vol. 18, No. 4, pp. 391-404, 2019.