GENETIC ALGORITHM - OPTIMIZED GATED RECURRENT UNIT (GRU) NETWORK FOR SEMANTIC WEB SERVICES CLASSIFICATION

S Sridevi[1*], G R Karpagam[2], B Vinoth Kumar[3]

[1,2]Department of Computer Science and Engineering, PSG College of Technology, 641004, Coimbatore, India

[3]Department of Information Technology, PSG College of Technology, 641004, Coimbatore, India

E-mail: sridevi92124@gmail.com[1*] (corresponding author), grk.cse@psgtech.ac.in[2], bvk.it@psgtech.ac.in[3]

DOI: https://doi.org/10.22452/mjcs.vol35no1.5

ABSTRACT

In the current era, as there is an abundant expansion of functionally similar web services, it becomes a prodigious issue for the web service discovery process. The service classification plays a significant role to greatly reduce the search space and retrieves the desirable service quickly and accurately. The classification is performed using the functional values. Recent research activities recommend RNN (Recurrent Neural Network) deep learning algorithms for efficient classification process. The state-of-the-art of GRU (Gated Recurrent Unit) one of the RNN model, provides a proficient classification process. However, the ratio of training and testing dataset, and hyperparameters namely neural network size, and batch size etc, affects the classification accuracy. The objective of the paper is to incorporate GRU model for efficient classification process. The novelty of the proposed model lies in implementing the GRU model for semantic web service classification. Furthermore, the genetic algorithm is used to predict the optimal ratio of training and testing dataset and optimal hidden neural Network units of GRU model in order to attain optimal classification. The experimental results exemplifies that the semantic web service classification is efficiently deliberated using the proposed GA-GRU model that outperforms the classification process as compared with the conventional semantic extraction using accuracy, precision, F-measure, recall and FDR (False Date Rate) rate.

Keywords: Web Service Classification, Sparse features, GRU, Genetic algorithm, Service discovery.

1.0 INTRODUCTION

By the rampant evolution of huge collections of splendid plentiful web services, web service classification has become one of the most significant tasks in the UDDI repository for automatic semantic web service discovery process [1], which aims to discover the factual list of web services for the given user request [2]. The systematic legitimate repository classifies the web services as a hinge to extract functional description, sparse features, and semantic relationship from WSDL or OWL-S file - a document that comprises the service provider's profile, model, and grounding information. It helps service users to pleasantly derive the exact web service from a massive amount of web services and provides interfaces to establish business collaboration between a service provider and service user via SOAP protocol [3]. For example, holiday trip booking assists to discover all mandatory web services like flight booking, spot visiting, cab booking, passport, and visa verification, lodging and boarding booking, etc, to satisfy the whole travel booking strategy.

A fundamental problem of web service classification is the learning of text semantics and sparse depictions that are encoded as continuous vectors by creating a high-dimensional semantic space projection. The ramification of text classification primarily hinges on mining semantic and compact features from given words via descriptive representational learning [4] [5] [6] [7] [8] [9]. Typically, there are two types of representational models for text classification based on proficient information and accessible knowledge. They are conventional machine learning characterization methods and current prominent deep-learning characterization methods. The conventional machine learning characterization models that train a classifier depending on semantics existence among textual knowledge and semantic lexicons [9]. However, the latest prominent deep learning models, automatically understand the existence of semantics among the text impeccably. Thus, the classification efficiency of deep learning models based on semantics is superior to that of machine learning approaches, which is complex for computing by conventional machine learning models.

The primary notion of performing classification in machine-learning representation models is to deploy text-based classifiers namely, Naive Bayes classifier, Smart Clustering models, ensemble learning methods, and Support Vector Machines [10] [11] [12] [13] [14], etc, towards extracting the semantic polarity of the given texts for domain-based classification. Jianxiao Liu et al [14] propose Naive Bayes Classifier for semantic web service classification (SWSC) that makes use of IOPE (Input, Output, Precondition, and Effect) features, that extract the linguistic among text by digging deeper among the given training features to perform classification process. The SVM model constructs the semantic relationship and semantic annotation among feature vectors to extract the relation of text it belongs to the specific domain. The functional semantic representation of the services acts as an effective solution to classify web services in domain-specific categories. However, these machine-learning representation methods did not address fine-grained semantic information like word order, word features, and contextual information of service description. Machine-learning representations only interpret the data from a qualitative point of view, and mathematical theory. Meanwhile, the aforementioned descriptive feature vectors are essential for efficient semantic web service classification.

In recent times it is been generally recognized that deep learning models have attained huge success in the text-based classification process. This is due to the ability of deep learning (DL) methods that automatically learns the semantics concealed behind the textual data without exploiting its strenuous features [15] [16]. Jason Brownlee et al proposed the Recurrent Neural Network (RNN) - DL model, which has the ability to automatically understand the semantic word context existing among text, which facilitates a cool prediction of textual classification from the given data [17]. Hochreiter et al proposed the LSTM network, which is an advanced model of the RNN network. It generally has good learning and extraction capability of sequential correlation of the texts using its functional gating mechanism [18]. Tai et al tried the LSTM model for sentimental classification and proved that the prominence and deep linguistic sequence extraction knowledge of the LSTM model, affords the effective classification predictions as compared with other ML classification models. Additionally, the author has also proposed a Tree-LSTM, which epitomizes the semantic lexical structure of text within its precise structure to perform effective classification [19]. The improved version of the LSTM model called the Bi_LSTM model is applied for web service classification [20], that is initially developed by Google for deep and wide prediction of data to perform classification and regression [21]. The wide contextual prediction of the Bi_LSTM model is used to memorize the sparse features and a deep neural component of Bi_LSTM automatically understands the unknown hiding interactions that exist among vital features via low-dimensional embedding. Thus, this model achieves an effective textual classification process.

However, the structure of LSTM has sadly lost the ability to extract the linguistic knowledge which presides among local text features, which is the essential property for an efficient classification process [22]. The structure of the GRU model is designed meritoriously to capture the effective reminiscence of linguistic knowledge and proficient categorization of unsolicited text, along with conserving the LSTM congenital efficiency for capturing the ground features of the text. In order to avoid the ignoring of context background features and sequential correlation information from the given text, it is essential to enrich a model that automatically understands the linguistic knowledge that exists among features. Muhammad Zulqarnain et al proposed the GRU model for word embedding text classification by accessing context information [23], correspondingly, Wenkuan Li et al proposed the Bi-GRU network to enrich a sufficient text representation using past and future contexts information for sentiment classification that achieves higher classification accuracy [24]. Yet, there is no research work is performed using the GRU model for semantic web service classification.

As inspired by the above research work of entrenching GRU for effective sentiment classification, the objective of the paper is to incorporate the GRU model for semantic web service classification. However, the performance of the GRU model mainly depends on hyperparameters namely neural network units, initial weights, a quantity of feeding training dataset, optimizer, etc. If a decision-maker ingrained fruitless hyperparameters, then the GRU model obtains local optimum results. With regard to this, some researchers have utilized metaheuristic algorithms to find effective hyperparameters for improving classification accuracy [25] [26] [27] [28] [29] [30]. In accordance with the previous research works, the paper proposed the GA+GRU for semantic web service classification.

The main contributions of the paper are summarized as follows:

1. The paper exploits conventional text lexicon and the mainstream existing system to design a new novel system for semantic web service classification. Meanwhile, the classification mechanism requires a strong capability to extract important semantic IOPE features and contextual interactions among the text. Accordingly, the GRU model generates sequential correlation information of IOPE text which is essential for web service classification tasks. Hence, the paper focuses on implementing the GRU model to enrich the semantic web service classification accuracy.

2. The initially designated GRU hyper-parameters significantly affect the accuracy level of web service classification. Usually, for the classification problem, the training dataset and size of neural network units impose consequential impacts over GRU prediction. For that, a Genetic Algorithm (GA) is incorporated to determine the most effective aforementioned hyper-parameters (window size, size of neural network units) for effective classification.

3. Extensive experiments were done on OWL-TC datasets with IOPE features along with domain labels to assess how good the GRU-GA model is at classifying semantic web services using accuracy, precision, F-measure, recall and FDR (False Date Rate) rate.. The experimental findings show that the proposed GRU-GA model significantly outperforms other RNN classification models.

The rest of the paper is organized as follows. Section 2 provides an insight into the Pre-processing of Web Service Description Document for classification. Section 3 discusses the working of the proposed work - GA-GRU, whereas Section 4 exemplified optimization of GA, and Section 5 discusses the working of the Genetic Algorithm. Whereas, the experimental evaluation and validation of the proposed methodology are discussed in Section 6. Finally, Section 7 gives the conclusion followed by references.

2.0 PRE-PROCESSING OF WEB SERVICE DESCRIPTION DOCUMENT

The overall context of the research is illustrated in Figure 1, that consists of 3 tasks namely, 1) determination of optimal classification training set in terms of window_size, 2) identification of optimum Neural Network (NN) units for computations, and 3) training web service classification set over GRU model. Initially, the IOPE is extracted as features from the Web Services description document at preprocessing stages whereas, the derived IOPE features are primarily converted to vector matrix. During GRU training, the derived IOPE feature vector matrices along with domain categories are trained over GRU components, thus GRU would extricate the generalization, semantic relation, word remembrance, context of words, sparse features from the classification feature vector. The optimal training data window_size and Neural Network units are determined by consistently executing the Genetic algorithm overrunning the GRU model. The joint execution of GA over the GRU model achieves proficient web service classification accuracy.

2.1 Preprocessing of Web Service Description Document to create of Word Vector

The whole functionality of Web Service (WS) is described by the web service description document (WSDD). The WSDD is the source for web service classification. As the description document contains lots of unwanted information, preprocessing is required to refine the WS document to extract explicit features of the web service. The example semantic web service document given for classification is illustrated in Table 1. The preprocessing of the WS document is attained by the following steps:

Step 1: The collection and acquirement of the web service description documents. The RapidMiner toolkit in python is used to extract the IOPE of web services such as web service name, input, output, precondition, and effect as separately from the web service document.

Step 2: Words are separated via space and punctuation is isolated from words. The tokenization method is carried out using the Python RapidMiner toolkit.

Step 3: Stop Word Removal: There are several invaluable texts and punctuation inscriptions in the description document in English like 'a', 'an', 'has', 'It' etc. Such words or phrases which had nope practical significance are termed stop words. The stop words are exploited using RapidMiner Toolkit.

Step 4: Stemming. A similar text in English has various expressions in different situations depending on the tense, phrases like "executing, stimulate, stimulating, stimulates", and so on. However, in reality, all these texts are representing the similar word "stimulate". The accuracy of similarity calculations will be diminished if these terms are perceived as different words. The stemming process has been performed using RapidMiner Toolkit.

Step 5: Document Representation: The data must be represented in the pattern, that the classification algorithm can recognize the Bag of Words (BOW). It labels the occurrence of words in the document.

Step 6: Dimension Reduction: It is not possible to feed all words in the documents as features. The algorithm is not able to compute such data. So, the best instructive features are given as inputs for the classification. RNN itself reduces embedded vectors into a lower dimension.

Step 7: Model Training: This is the main phase for classification. Here, a certain portion of the text is selected from the dataset, applying training over the learning set, and producing the model.

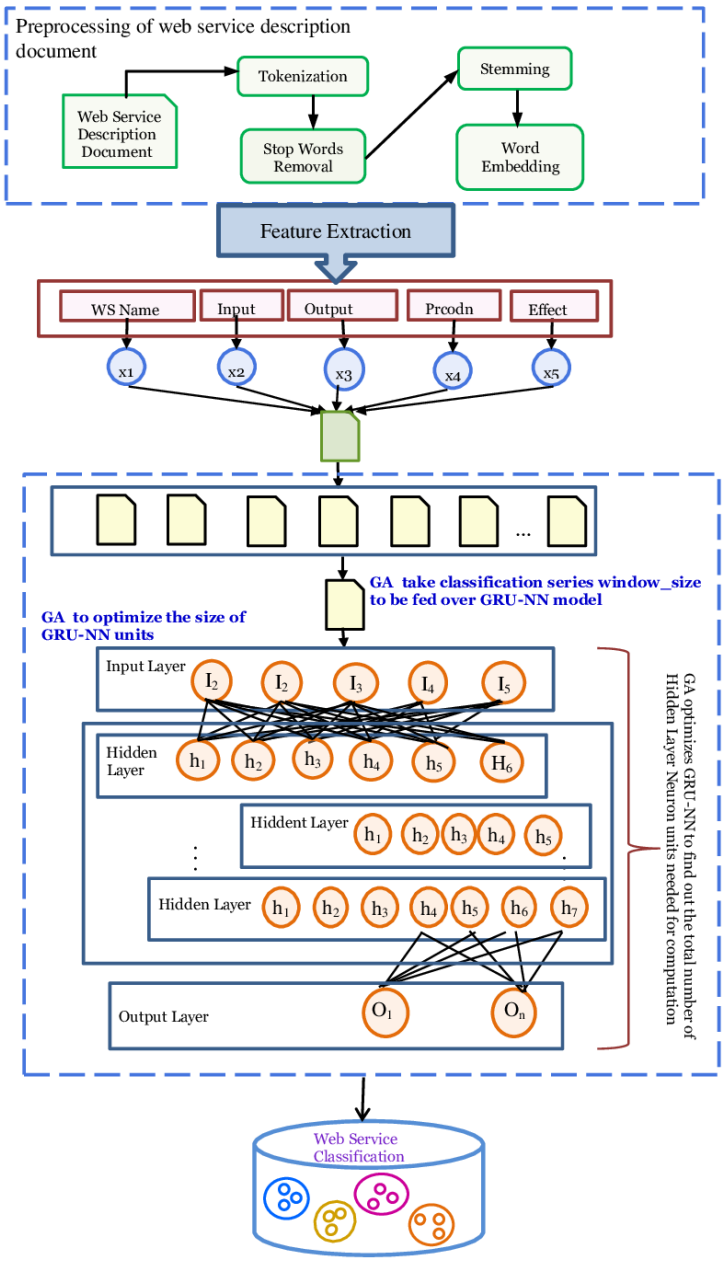

Fig 1: Overall Framework of Semantic Web Service Classification

Table 1: Example semantic web service document for classification

service name	hasinput	hasinput	hasinput	hasinput	hasoutput	hasoutput	precondition	effect
tradition al food service	South indianfood	North indianfood	Traditional sweets	chennai	Tastyfood	doordelivery	cash only, card accepted	order confrom
zillanq food service	South indianfood	North indianfood	chinesefood	all over country	Your foodcourt	Enjoy your food	cash only, card accepted	book chair
fast food sevice	italian	chinese	indian	New jersy	chats	take home	cash only	book chair
.	.	.	.	.	.	.	.	.
Web-stat service	Find location traffic	time-measure	map images	-	matching destinator location	enable traffic less route	location on	Traffic status detected enabled traffic less route

3.0 THE PROPOSED SEMANTIC WEB SERVICE CLASSIFICATION MODEL

The proposed model is described as follows

3.1 Training of the GRU Network

The deep learning GRU Neural Network / GRU model extracts the features autonomously from the prepared training dataset. It captures and generalizes the features in order to determine a deeper and more abstract feature matrixto mine-wide components of WS categories. The efficiency of the GRU model for classification depends on the weights of the update and reset gate as well as the optimization of hyper-parameters [23].

GRU-RNN Network

GRU Network autonomously learns feature vectors from a given training set. To find deep and more abstract WS features, GRU captures and generalizes the features to mitigate the impact of small components. Therefore, the GRU model can able to generalize the composition of invisible WS features by low-dimensional dense embedding for sparse vectors. As the WS features vector is sparse, GRU combined with embedded characteristics perform encapsulation to mine deep feature vector. Besides, it also identifies dependencies between word order and word context. GRU uses two gates 1) update gate and 2) reset gate to classify web services. The amount of conceded information from the previous activation state to a new candidate state is tied and computed by the update gate. The flow of information and the amount of unsolicited information to be forgotten from the previous activation with a new candidate state is handled by the reset gate.

For example, some semantic similarities between the web service vectors are extracted from traditional food service $(T.H.S)$, *zillaq food service* $(Z.H.S)$, *and fast food*$(F.F)$ *as,*

1. $vector\ (T.H.S) - vector\ (Z.H.S) + vector\ (indian\ food) = indian\ food$
2. $vector\ (T.H.S) - vector\ (F.F) + vector\ (chinese\ food) = chinese\ food$
3. $vector(food) - vector\ (tasty\ food) + vector\ (south\ indian\ food) = vector\ (T.F.S)$

.

.

$n.\ vector(doordelivery) - vector\ (cash\ payment) + vector\ (south\ indian\ food) = vector\ (T.F.S)$

Based on semantic vector computation, the update gate and reset gate control the flow of information to make the decisions. The typical GRU-RNN model is shown in Figure 2.

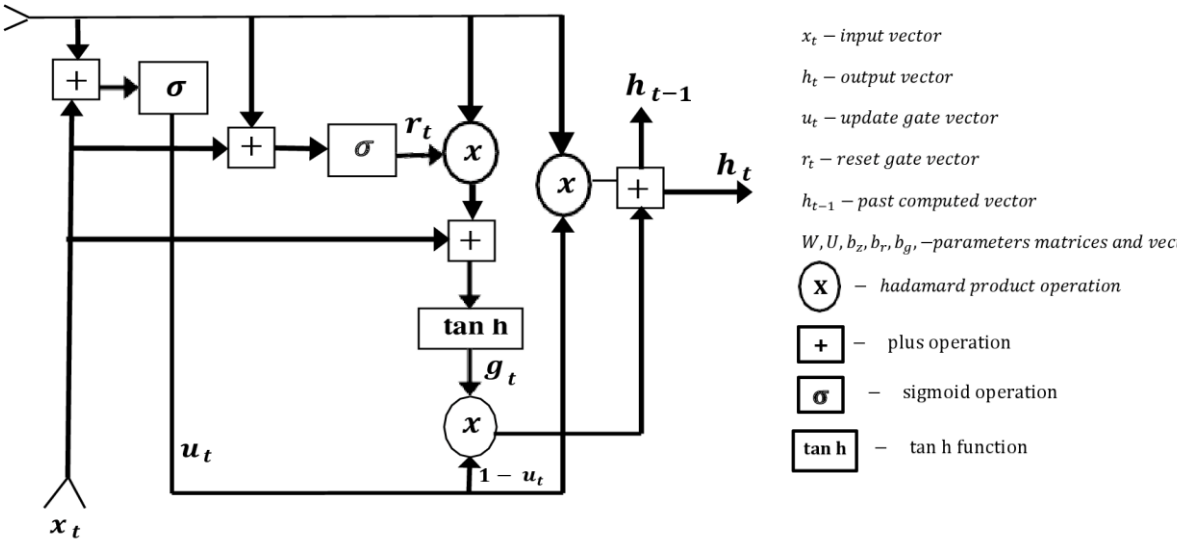

Fig. 2: Structure of GRU-RNN network

1. Update gate

The update gate determines how much semantic similarity features are needed to pass for computation. This model extracts all semantic similarity features between the web services and provides the important features required for classification. Update gate is computed as in equation 1

$$Update\ gate:\ u_t = \sigma\left(W_z x_t + U_z h_{t-1} + b_z\right) \quad - \quad (1)$$

The vector results are computed with a sigmoid activation function to squash the result between 0 and 1.

2. Reset gate

This gate decides how much semantic relation between various web services to be forgotten. It is computed as in equation 2

$$Reset\ gate:\ r_t = \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right) \quad - \quad (2)$$

Reset gate results are computed with sigmoid activation function and squash the result between 0 and 1.

3. Current memory content

A current memory state remembers the past computed semantic similarly between the various services. It is computed by equation 3.

$$Candidate\ gate:\ g_t = tanh\left(W_h x_t + r_t \odot U h_{t-1} + b_g\right) \quad - \quad (3)$$

4. Final memory

h_t is the final vector result that would be passed down to the network. For final vector h_t computation, it needs an update gate and reset gate to determine the semantic similarity to be passed from the previous state and semantics to be ignored from the candidate state. It is determined by u_t and r_t . If u_t vector values are close to 1 decide to preserve semantic information. Subsequently, if r_t vector values are close to 0 which determined to ignore semantic content. It is computed by equation 4.

$$Final\ output:\ h_t = u_t \odot h_{t-1} + (1 - u_t) \odot g_t) \quad - \quad (4)$$

Final GRU-NN classification Layer

In GRU-NN, softmax regression is often instigated as a final layer for binary and multiclass text classification.The computation is fast and provides a probabilistic description outcome. The classification layer polishes fluctuations followed by a softmax activation feature to measure predictive possibilities of all categories which are attained by equation 5,

$$y = \frac{\exp\left(W_o^T S^* + b_o\right)}{\sum_{i=1}^{n} \exp\left(W_o^T S^* + b_o\right)} - (5)$$

Where Y is the text's expected distribution, W_O^T and b_o are the learning parameters of the softmax classifier. As a result, the GRU model can generalize imperceptible Web service features of the given web service document by entrenching low-dimensional sparse factors to mine the dense factors which are hiding behind the vectors of the given document.

4.0 OPTIMIZATION OF CLASSIFICATION SERIES WINDOW_SIZE AND GRU-NN UNITS USING GENETIC ALGORITHM

The optimization segment comprises of two phases, which are as follows:

1. First task: To design the appropriate GRU_NN hidden layer units and classification series window_size. The GRU_NN designed for this classification experiment has one sequential input layer followed by one hidden layer and one sequential output layer.

2. Second task: To fix other GRU–RNN factors as follows: the hyperbolic tangent function has been used as an activation function between input units and hidden units. The hyperbolic tangent function is a scaled sigmoid function that extracts the vector values into a range of -1 and 1. The output unit's activation function has been labeled as a linear function. The initial weights of the network are set randomly, and the weight has been attuned via an "Adam" optimizer, which is known for its clarity, simplicity, and computing efficiency. This approach is well suited to problems involving huge amounts of data and parameters, as well as non-stationary challenges caused by noise and sparse gradients.

As previously stated, GA, which is one of the evolutionary search algorithms used to examine the optimum classification series window size and hidden units of the GRU-NN network. In the second stage, various randomly generated sizes of classification series window size are computed with randomly generated GRU-NN hidden units are computed to evaluate the fitness of GA.

Estimation of Classification series window size

Various sizes of classification series window_size N^{WS^*} are designed to predict the ratio of dataset given for GA computation is shown in equation 6 and equation 7.

$$t_D = \arg(ratio)\, P\,(change - D_S|\, i{:}j) \quad - \quad (6)$$

$$N^{WS^*} = N^*(t_D) \quad - \quad (7)$$

Where, N^{WS^*}- Optimal window size, 'P' - the probability of splitting the dataset, $'i'$ - training set $'j'$ - validation set, t_D – splitting ratio get from GA, N^* - optimal classification ratios given for the model. The given dataset has 1083 instances with 10 attributes. Here, 1083 is divided into three segments to perform computation in GA as a training set, validation set, and testing set. The dimensionality of attributes is defined by ten instances. Each class instance should take ten dimensionality or features as input and compute to predict the output label domain category. This automatic computation is performed to avoid the existence of the overfitting and underfitting problem. The length of window size is represented by binary encoded bits. The chromosome for classification series window_size is represented as in Figure 3. The sample semantic web service classification series window_size computation is illustrated in Table 2, and Table 3, and Figure 4 and Figure 5 illustrate the bit-encoding values of classification series window_size of Table 2 and Table 3.

0	1	1	1	1	0	1	1	0	1

Fig. 3: Chromosome representation of classification series_window size

Table 2: Classification series window_size

S.No	service name	hasinput	hasinput	hasinput	hasinput	hasoutput	hasoutput	precondition	effect
1	traditional food service	South indian food	North indian food	Traditional sweets	chennai	Tasty food	doordelivery	cash only, card accepted	order confrom
2	Zillanq food service	South indian food	North indian food	chinese food	all over country	Your food court	njoy your food	cash only, card accepted	book chair
3	fast food sevice	italian	chinese	indian	New jersy	chats	take home	cash only	book chair
4	delighted tea coffee service	tea	coffee	wayanad	all over country	relaxing hot drinks	doordelivery	book online	order confrom
. .	. .	. .	. .	. .	. .	. .	. .	. .	. .
714	Web-stat service	Find location traffic	time-measure	map images	-	matching destinator location	enable traffic less route	location on	Traffic status detected enabled traffic less route

1	0	1	1	0	0	0	1	0	1

Fig. 4: chromosome representation of classification series window size -714

Table 3: Classification series window_size

S.No	service name	hasinput	hasinput	hasinput	hasinput	hasoutput	hasoutput	precondition	effect
1	traditional food service	south indian food	north indian food	traditional sweets	chennai	tasty food	doordelivery	Cash only, card accepted	order conform
2	MIR service	types	physician	geograph ical location	DP service	MIR scan	laboratory report	Book appointment	appointment booked
3	fast food service	italian	chinese	indian	new jersy	chats	take home	cash only	chair booked
4	Saarlandhos pital service	physician	working hours	specialized area	chennai	book appointment ent	-	credit and debit cards only	booked appointment
. .	. .	. .	. .	. .	. .	. .	. .	. .	. .
511	munchen university	engineering	faculties	geograph ical location	environment	admission details	counseling from submission	12 th score above 1100	conform submission

0	1	1	1	1	0	1	1	0	1

Fig. 5: chromosome representation of classification series_window size -831

As the 1x10 classification series window_size is randomly fit with randomly generated GRU-NN units. The chromosome of NN is represented in Figure 6.

0	1	1	1	1	1

Fig. 6: chromosome representation of GRU-NN units

These aforementioned randomly generated classification series window_size are computed with randomly generated GRU-NN units to evaluate the fitness of GA. Before the genetic operators begin to explore the search space, the populations of candidate solutions are initialized with random values. The GA is evaluated by the following steps:

1. Chromosome Initialization

The mutual chromosome set of classification series window-size and GRU-NN units is illustrated as a 1x16 matrix. Thus, each generated chromosome has 16 genes. Each 1x16 chromosome is divided into 1x10 and 1x6 subtables as shown in Figure 7. From the left, the first 1x10 genes in the 1x16 matrix table signify the classification series window_size, and the second 1x6 matrix in the 1x16 matrix denotes the size of GRU – NN hidden layer units needed for performing classification. The preliminary chromosome is generated as in Algorithm 1.

Algorithm 1: Initialization algorithm

$Input$: $N\ number\ of\ initial\ chromosomes$
$Output$: N number of 1x16 chromosomes

$i)$ $Each\ 1\ x\ 16\ chromosome\ is\ divided\ into\ 1x10\ and\ 1x6\ subtables$
$ii)$ $Get\ the\ number\ of\ initial\ chromosomes$
$iii)$ $Randomly\ generate\ the\ individuals\ of\ the\ initial\ population$
$iv)$ $Find\ duplicates\ and\ replace\ new\ ones$

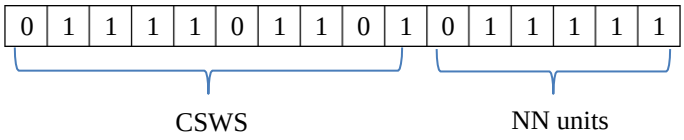

Fig 7 : Chromosome Initialization

2. Fitness Evaluation

Initially, the fitness function of 1x10 and 1x6 subtables are evaluated separately, and afterward, the entire 1x16 matrix chromosome fitness is evaluated by the validation accuracy score, which is shown in equation 8. Also, False Date Rate (FDR) is considered to identify the falsified prediction about the computation of specific chromosomes which is shown in equation 9. The fitness function is evaluated as in Algorithm 2.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad - \quad (8)$$

$$FDR = \frac{FP}{FP + TN} \quad - \quad (9)$$

where, TP-number of true positives, FP-number of false positives, FN-number of false negatives, TN-number of true negatives

Algorithm 2: Fitness Evaluation

$Input$: $Randomly\ generated\ 1x16\ chromosomes$
$Output$: $calculate\ the\ fitness\ of\ the\ 1x16\ chromosomes$

$i)$ $The\ initial\ population\ chromosomes\ are\ given\ for\ fitness\ evaluation$
$ii)$ $A\ randomly\ generated\ 1x10\ subtable\ is\ computed\ with\ each\ randomly\ generated\ 1x6\ subtable$
$iii)$ $For\ the\ entire\ 1x16\ matrix, the\ fitness\ is\ evaluated$
$iv)$ $Accuracy\ is\ evaluated\ to\ predict\ the\ fitness\ of\ a\ chromosome$
$v)$ $The\ chromosomes\ having\ high\ accuracy\ scores\ are\ selected\ as\ a\ parent\ for\ crossover$

3. Selection

Superior offsprings are selected based on the Roulette Wheel Tournament selection operator. The Tournament selection operator can traverse through the entire K population set and pick the best-fitted chromosomes for the matting pool as shown in Figure 8. The chromosomes are selected as in Algorithm 3.

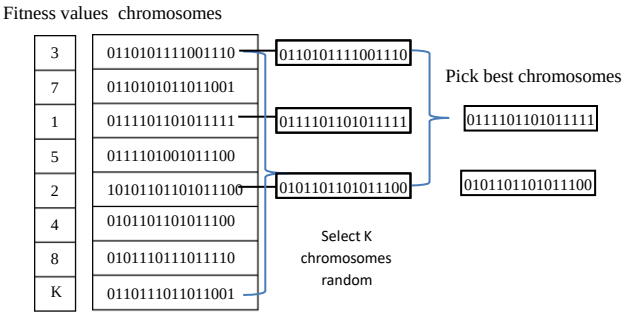

Fig 8 : Tournament selection

Algorithm 3: Roulette wheel Tournament selection

Input: *A population of size K with their fitnessC*
Output: *A mating pool of size* K_P $(K_P < K)$

i) *select* K_P *individuals at random* $K_m < K$
ii) *From* K_m *individuals select the chromosomes with high fitness values as the winner*
iii) *Add the winner as mating pool*
iv) *Repeat the above steps until the mating pool contains* K_P *individuals*

4. Uniform Crossover

The uniform crossover is implemented to generate new offspring. The uniform crossover is applied separately for 1x10 and 1x6 subtables. The generation of offsprings using uniform crossover is shown in Figure 9. The crossover is achieved as in Algorithm 4.

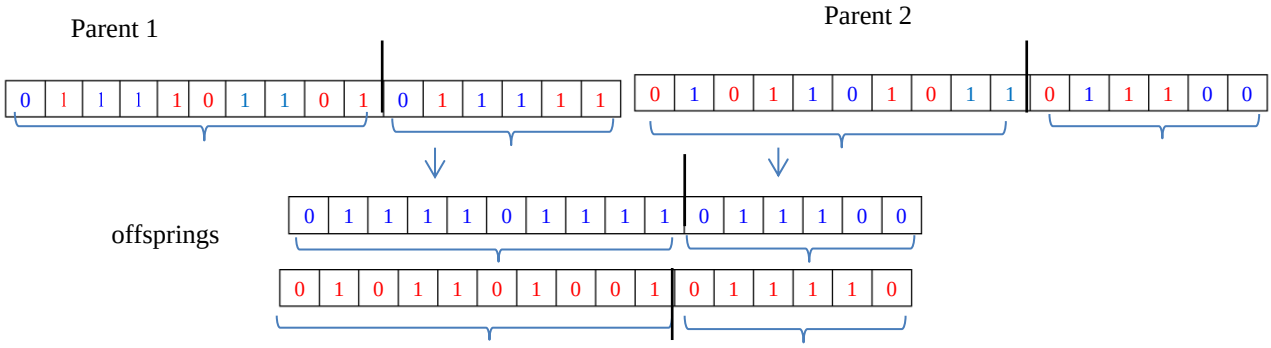

Fig. 9: Uniform Crossover

5. Bit inversion Mutation

Mutation assesses to search for new possible solutions by changing the gene values in the chromosome randomly. Here, Bit inversion mutation is performed on 1x10 and 1x6 subtables, by changing the gene value of newly generated offspring to obtain optimal fit chromosome. The mutated offspring is shown in Figure 10. The bit inversion mutation is performed as given in Algorithm 5.

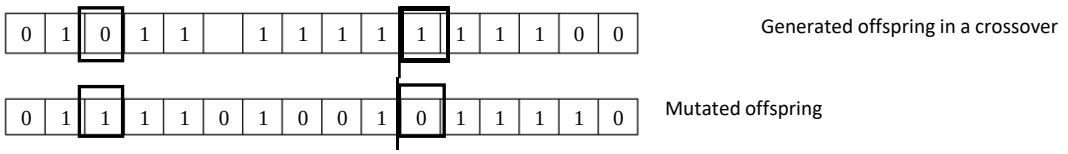

Fig. 10: Bit Inversion Mutation

Algorithm 4: Bit Inversion mutation

$Input: Superior\ Parent\ chromosomes$
$Output: Offsprings$

$i)$ $All\ superior\ chromosomes\ from\ the\ crossover\ pool\ are\ sorted\ between\ 1\ to\ P$
$ii)$ $Select\ 'p'\ a\ random\ number\ between\ 1\ and\ P$
$iii)$ $Perform\ bit\ inversion\ mutation\ of\ randomly\ selected\ individual\ of\ two\ subtables$
$iv)$ $Execute\ permutation\ over\ selected\ pairs$
$v)$ $Generate\ optimal\ offsprings$

5.0 WORKING OF GENETIC ALGORITHM

Initially, the chromosomes are randomly generated as a 1x16 matrix. Each chromosome is divided into two 1x10 and 1x6 subtables. The fitness function of the subtables of the chromosome is evaluated by classification series size and NN units, hitherto, the fitness function of the entire chromosome is evaluated by accuracy score. The chromosomes that have a high accuracy score are considered highly fitted chromosomes. The highly fitted chromosomes are selected to attain uniform crossover operation in order to generate offspring. A Bit inversion mutation is computed to generate a superior offspring from the newly generated offspring. Fitness function is evaluated for each newly generated offspring. Consequently, the newly generated offspring and parent chromosomes are considered as the current population and start to generate new offspring. This process is repeated until the desired number of generations has been reached. The chromosome with a high fitness value is selected as an optimal chromosome.

6.0 EXPERIMENTAL EVALUATION

The experiment is evaluated from the OWLS_TC data set which has been taken from the URL - http://projects.semweb central.org/projects /owls-tc/. It includes 1083 WSDL service files which have the collections for the following nine domains: geography-60, education-286, weapon-40, communication-59, medical care-73, economy-395, food-34, stimulation-16, and travel-197. The reasoning tool Mindswap OWL-S API has been used to determine the semantic relationship between concepts. Here, all the 1083 web service files are selected for experimental analysis and comparison. For GA computation, the values for hyperparameters optimization are accomplished based on the motivation of the GA-LSTM paper proposed by Hyejung Chung for stock market prediction. The crossover probability is fixed between 0.3 to 0.9 to evaluate the performance of the proposed GA-GRU classifier in various crossover probability rates. The parametric values for genetic algorithm computation are represented in Table 2.

A. Experiment Settings

During the experiment, all 1083 services are selected and further five technical indicators are extracted as IOPE (Input, Output, Precondition, Effect) and domain category utilized as input variables by reviewing prior research works and domain experts. The prepared web service dataset is given for automatic computation to predict the training set and testing set based on validation accuracy score. The high-fitted chromosomes are formerly fed into the GRU computation model in Keras Python Sklearn. In the GRU model, some significant parameters has been formally fixed as: Batch_size = 32, Num_epochs = 50, optimizer = adam,dropout rate = 0.2.

Table 2: GA Parameters

Parameters	Experiment settings
Initial Population	70
Probability of Crossover	0.3 – 0.9
Probability of Mutation	0.13
Number of Generations	100
Window Size	1-1083
GRU-NN Units	1-35

B. Evaluation Metrics

In this experiment, the 'closed test' principle is adopted to evaluate the performance of the proposed GA-GRU model. The standard classification result of the web service document is formulated as $SWSC = \{C_1, C_2, C_3, \dots, C_k\}$, and the experimental classification result as $ESWSC = \{EC_1, EC_2, EC_3, \dots, EC_k\}$. The precision, recall, f-measure, and accuracy of the ESWSC are calculated as described in equations 9 to 12 and compared with SWSC which are described as follows:

$$Precision = \frac{TP}{predicted\ yes} \quad - \quad (9)$$

$$Recall = \frac{TP}{actual\ yes} \quad - \quad (10)$$

$$F - Measure = \frac{2 * precision * recall}{precision + recall} \quad - \quad (11)$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad - \quad (12)$$

$where, TP - number\ of\ true\ positives, FP - number\ of\ false\ negatives, FN - number\ of\ false\ negatives, TN - number\ of\ true\ negatives$

Experimental Results and Analysis

In this section, we compare the efficiency of the GA-GRU model in various crossover probability ranges in terms of Accuracy, In addition, classification efficiency is compared with other RNN Baseline models in terms of precision, recall, f-measure, and accuracy. The experiments are assessed over the web service classification dataset in the GA-GRU model by setting different combinations of crossover probability and mutation probability rate which is illustrated in Table 2. Table 3 demonstrates the optimal output of evaluated experiments as fitting the various combinations of crossover probability rate and mutation rate and Figure 11 and Figure 12 pictorially illustrate the optimal outcome of the GA-GRU model in terms of accuracy and FDR rate. In the Figure, the horizontal coordinate value denotes the crossover probability rate and the vertical coordinate value denotes the corresponding evaluation metric. While Figures 13-16 exemplify the comparison of classification performance of various RNN models namely LSTM, GRU, Bi_LSTM, Bi_GRU, and GA-GRU models. In the figures, the horizontal coordinate denotes the WS domain classes, and the vertical coordinate denotes the corresponding evaluation metric. The comparison reveals that our proposed GA-GRU model significantly increases the classification performance in terms of recall, precision, F- measure, and accuracy, which shows that the proposed model outperforms all other RNN baseline methods in all cases.

Table 3 : Optimal output of different crossover probability computation

Optimal Set	Total no of parameters	WindowSize	Num of Units	Accuracy	FDR rate
C.P =0.3 M.P=0.25	1314	842	27	79.67	0.083
C.P =0.4 M.P=0.3	1097	749	31	76.27	0.078
C.P =0.5 M.P=0.15	1238	673	27	86.9	0.074
C.P =0.6 M.P=0.25	1295	732	31	75.89	0.093
C.P =0.7 M.P=0.15	1196	836	28	92.34	0.046
C.P =0.8 M.P=0.2	1143	746	21	87.61	0.061
C.P =0.9 M.P=0.1	1221	539	32	85.61	0.064

6.1 Comparison of GA-GRU efficiency at different crossover probability values

The optimal output of different crossover probability range is illustrated in Table 3, which compares and analyzes the proposed GA-GRU model computation efficiency at various crossover probability rates in terms of accuracy. The performance accuracy is illustrated in Figure 11. From Figure 11, it can understand that the web service classification accuracy is maintained at more than 90% at the crossover probability range of 0.7. This shows that the proposed GA-GRU computation efficiency is higher at the crossover probability range of 0.7 and the mutation rate as 0.13. Hence, crossover probability rate-0.7, and Mutation rate-0.13 are fixed for the proposed GA-GRU web service classification model computation.

Comparison of GA-GRU classifier accuracy at a various crossover probability value

The accuracy of web service classification is higher at crossover probability rate-0.7. Correspondingly, it has been seen that accuracy of crossover probability rate-0.7 is an 18.5% improvement over the crossover probability rate of 0.3, 23.69% improvement over the crossover probability rate of 0.4, 8.6% improvement over the crossover probability rate of 0.5, 24.3% improvement over the crossover probability rate of 0.6, 4.78% improvement over the crossover probability rate of 0.8, and 7.7% improvement over the crossover probability rate of 0.9. The reason for this effective classification of crossover probability rate - 0.7 is to generate operative effective offsprings for GA computations. From Figure 8, it has been proved that the crossover probability rate 0.7 can build the most effective GRU model, and it would lay the foundation to build an efficient GA-GRU model for semantic web service classification.

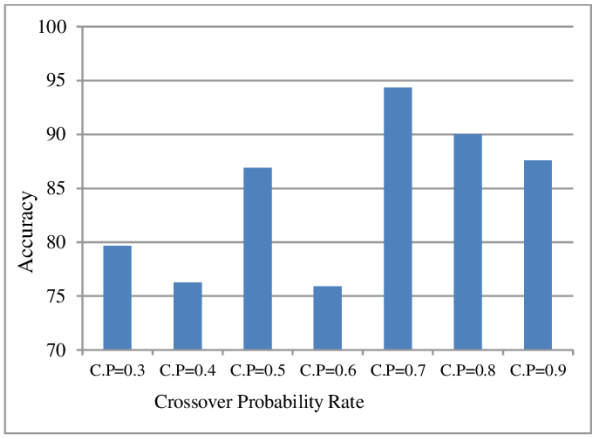

Fig. 11: Accuracy comparison of different crossover probability rates for GA computation

Comparison of FDR (False Data Rate) of the classifier at a various crossover probability value

Here, the web service classification of the proposed GA-GRU model is compared in terms of the FDR rate. The pictorial result is illustrated in Figure 12. It shows that the FDR rate of crossover probability rate-0.7 is 57.36% lower over the crossover probability rate of 0.3, 51.61% lower over the crossover probability rate of 0.4, 46.67% lower over the crossover probability rate of 0.5, 67.62% lower over the crossover probability rate of 0.6, 26.41% lower over the crossover probability rate of 0.8, and 32.72% lower over the crossover probability rate of 0.9. It ensures that crossover probability rate-0.7 affords good in-fit over GA-GRU computation, associated with low error measures, and avoids overfitting of the model.

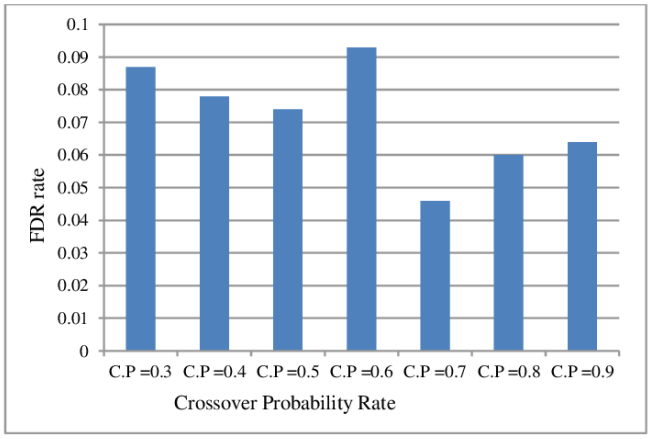

Fig 12 : FDR rate comparison of different crossover probability rates for web service classification

6.2 Semantic Web service Classification Prediction

Example: The newly entered web service - Chettinadu food Service defined in Table 4,

Table 4: Web service request to add the new service to the respective domain

service name	hasinput	hasinput	hasinput	hasinput	hasoutput	precondition	effect
Chettinadu food service	South indian food	Non-veg	Chettinadu special bakes	Allover country	Chettinadu special food	cash only,card accepted	order conform

The GA-GRU model computes and predicts the semantic similarity among web services using the computation of vector is shown as follows,

1. $vector\ (T.H.S) - vector\ (C.F.S) + vector\ (south\ indian\ food) = south\ indian\ food$
2. $vector\ (Z.F.S) - vector\ (C.F.S) + vector\ (south\ indian) = south\ indian\ food$
3. $vector(C.F.S) + vector\ (F.F) + vector\ (bakes) = food$

.

.

$n.\ vector(C.F.S) + vector\ (T.H.S) + vector\ (tasty\ food) = food$

As, computing the semantic similarity of new web service is much more matched in the food domain, the Chettinadu food Service is added to food domain. The new updated web service classification list is shown in Table 5.

Table 5: Updated web service list

Food Domain									
S.No	**servicename**	**hasinput**	**hasinput**	**hasinput**	**hasinput**	**hasoutput**	**hasoutput**	**precondition**	**effect**
1	Traditional food service	South indian food	North indian food	Traditional sweets	chennai	Tasty food	Tasty indian food	doordelivery card accepted	order confrom
2	zillanq food service	South indian food	North indian food	chinesefood	all over country	Your food court	Enjoy your food	cash only, card accepted	book chair
3	fast food sevice	italian	chinese	indian	New jersy	chats	take home	cash only	book chair
4	delightedtea coffee service	tea	coffee	wayanad	all over country	relaxing hot drinks	doordelivery	book online	order conform
35	**Chettinadu food service**	**South indian food**	**Non-veg**	**Chettinadu special bakes**	**Allover country**	**Chettinadu special food**	**-**	**ash only, card accepted**	**order conform**

6.3 Baseline Methods

The following state-of-the-art existing methodologies were chosen for the experiment

LSTM: An important computation is hinged on extracting semantic similarities among classification sets. The LSTM model extracts these semantics from feature vector matrices from the given IOPE data and extensively accomplishes the prediction matrix to detect that particular WS domain label it belongs to. The LSTM simply utilizes the WS document's historical background knowledge, i.e. the pre-contrive information, to identify the category Web service belongs too.

Bi_LSTM: the working of input and output layers of the Bi_LSTM model is similar to that of the LSTM model. Nonetheless, the Bi- LSTM has two parallel layers one layer for computing forward direction and another layer for extracting the missing information by traveling backward direction. This computation not only mines the linguistic semantic background knowledge from the WSDD (i.e. pre-contrive information) rather it deliberates potential contextual information of the WSDD to perform classification.

GRU: the GRU model is an enhanced version of the LSTM model in terms of its network structure and performance. However, it does not address the LSTM's inherent flaw in collecting linguistic knowledge from the given local textual information. When dealing with original IOPE text, it is critical to extract sufficient text representation. The enriched GRU network is designed in a way to extract the semantics from the given text representation. It avoids the consideration of unwanted sequence correlation and inhabitant the undesirable contextual background characteristics.

Bi_GRU: the input and output of the Bi_GRU model are similar to those of the GRU model. It has two parallel layers of GRU networks in both forward and backward directions. That is, it not only extracts the context information from the WSDD (i.e. pre-contrive information), but it also extracts the semantic, and long terms relationship between word context from the WSDD (i.e. succeeding information) to perform classification.

GA-GRU: In GA-GRU neural network model, the GA performs optimization of hyper-parameters, which affects the performance of the GRU neural network. It combines the benefits of the GA model to optimize GRU-NN to enhance classification performance.

The experimental outcomes of web service classification over various Baseline RNN models are models is shown from Figure 13 to Figure 16. The average performance of the domain-based WSC process in the GA-GRU network shows higher performance as compared with the other four models evaluated under the same category. The precision rates of the GA-GRU model are 17.25% higher than the LSTM model, 15.5% higher than the GRU model, 7.13% higher than the Bi_LSTM model, and 4.27% higher than the Bi_GRU model. The reason for this effective classification is by effectual sensible mining of semantic information, which avoids

falsify classification. The optimized neural network units and training dataset are computed concurrently to attain a higher precision rate that is illustrated in Figure 13.

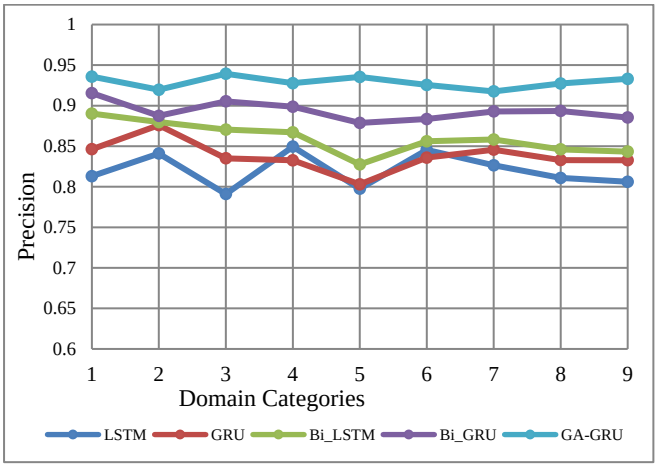

Fig. 13: Precision comparison of existing DL models
and proposed GA-GRU model for SWSC process

The recall of the GA-GRU model has a sharp improvement of nearly 16.25% as compared to the LSTM model, 14.5% enhancement over the GRU model, 8.82% enhancement over the Bi_LSTM model, and 4.07% improvement over the Bi_GRU model. It is attained by properly grasping the semantic and context categories effectively, that evades incorrect classification of positive classes from the given dataset as compared with other RNN models. Thus, the recall metrics of SWSC of the GA-GRU model are greatly improved which is illustrated in Figure 14.

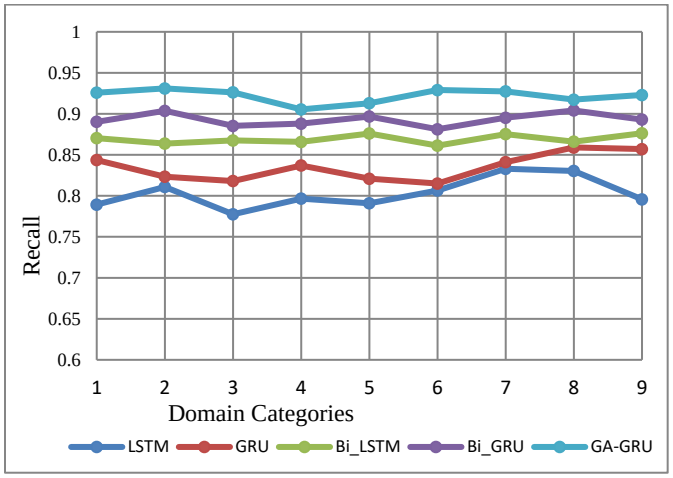

Fig. 14: Recall comparison of existing DL models
and proposed GA-GRU model for SWSC process

Similarly to the average precision and recall rates, the suggested GA- GRU model's average f-measure rate of WS domain categorization is greater than the other four RNN baseline models. The domain-based classification's precision and recall values in the testing set are greater than the domain-based classification's precision and recall values in the validation set. The F-Measure rate of the GA-GRU model achieves higher performance, due to its enhancement over achieving higher precision and recall rates. The F-Measure of the GA-GRU model is 16.09% is higher as compared with the LSTM model, 11.64% is higher over the GRU model, 10.5% is higher over the Bi_LSTM model, and 5% higher over the Bi_GRU model that is illustrated in Figure 15.

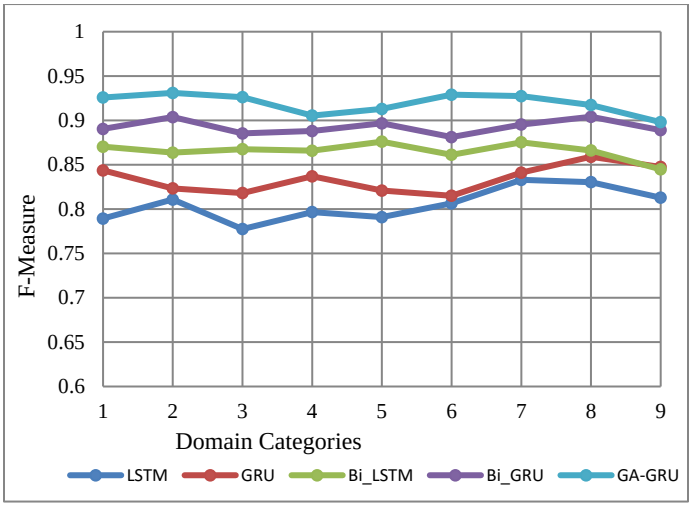

Fig. 15: F-Measure comparison of existing DL models and proposed GA-GRU model for SWSC process

The GA-GRU model achieves higher accuracy for the SWSC process under a similar category to that of the other four models. The classification accuracy of the proposed GA-GRU model has 17% progression over the LSTM model, 15% progression over GRU, 9.19% progression over the Bi_LSTM model, and 4.12% progression over the Bi_GRU model. The reason for this effective classification by the GA-GRU model is by reducing the occurrence of overfitting and underfitting problems that arise in classification sets by computing required data on its own. It also has the advantage of easy adding of input gate if needed. Thus, the automatic optimization of GA-GRU neural network units along with the optimal selection of dataset are mutually computed together to achieve a higher accuracy that is shown in Figure 16.

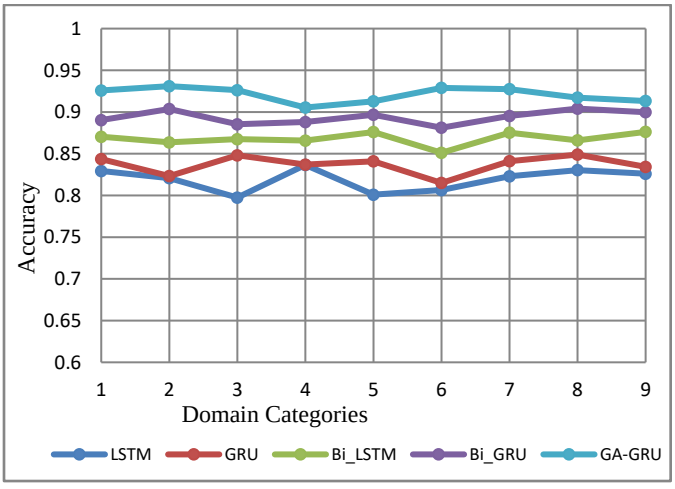

Fig. 16: Accuracy comparison of existing DL models and proposed GA-GRU model for SWSC process

7.0 CONCLUSION

In the service-oriented computing era, the classification of web services is an important task to enhance the discovery process which has not too been solved yet. This paper uses a SWSC method based on the GA-GRU model. The deeper generalized feature extraction by the GRU model, which captures the semantics among words effectively, and also has the capability to automatically understand the linguistic knowledge relation among feature vectors from the given web service description document. The optimization over GRU Neural Network is assessed by the Genetic Algorithm (GA) to enhance interpretability among Neural Networks. The comparative experiments conducted on OWL_TC dataset validate the efficiency of the proposed method and illustrate that the proposed approach significantly improves the quality of the classification of semantic web services in terms of precision, recall, F-Measure, and accuracy.

REFERENCES

[1] Boualem Benatallah, "Mohand-Said Hacid, Alain Leger, Christophe Rey, & Farouk Toumani". *On Automating Web services Discovery*, VLDB Journal, *Vol.* 14, 2005, pp. 84–96.

[2] UDDI, "The UDDI technical white paper". Retrived from: *http://www.uddi.org/pubs/Iru_ UDDI_Technical_White_Paper.pdf*, on September 1 2020.

[3] F. Curbera et al., "Unraveling the Web Services Web: an Introduction to SOAP, WSDL, and UDDI". *Internet Computing*, *Vol. 6*, 2002, pp. 86-93.

[4] Katakis. I et al., N, "On the combination of textual and semantic descriptions for automated semantic web service classification". *IFIP International Conference on Artificial Intelligence Applications and Innovations*, Springer, 2009, pp. 95–104.

[5] Salton. G et al., "Term-weighting approaches in automatic text retrieval. *Information Process Management*, Vol. 24 No. 5, 1988, pp. 513– 523.

[6] Salton. G et al., "A vector space model for automatic indexing". *Communication. ACM., Vol.* 18 No. 11, 1975, pp. 613–620.

[7] Xinghua lu et al., "Enhancing Text Categorization with Semantic-enriched Representation and Training Data Augmentation". *Retrieved form - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1561790/pdf/ 526.06001174.pdf*"., on August 2020.

[8] Jenq-Haur Wang et al., "An LSTM Approach to Short Text Sentiment Classification with Word Embeddings". *The 2018 Conference on Computational Linguistics and Speech Processing , ROCLING* 2018., 2018, pp. 214-223.

[9] Min Shi et al, "WE-LDA: A Word Embeddings Augmented LDA Model for Web Services Clustering". *IEEE International Conference on Web Services (ICWS)*, 2017.

[10] Yang. J et al., "Semi-automatic algorithm based on web service classification". *Advance Science Technology Lett", Vol.* 53, 2014, pp. 88–91.

[11] S. Sowmya Kamath et al., "Semantics-based Web service classification using morphological analysis and ensemble learning techniques". *International Journal of Data Science and Analytics., Vol.* 2, 2016, pp. 61–74.

[12] Y. J. Li et al., "Web Service Classification Based on Automatic Semantic Annotation and Ensemble Learning". *IEEE 26th International Parallel and Distributed Processing Symposium Workshops & Ph.D. Forum.*, 2012, pp. 2274-2279.

[13] Jianxiao Liu et al., "An Approach of Semantic Web Service Classification Based on Naive Bayes". *2016 IEEE International Conference on Services Computing*, September 2016.

[14] Pengfei Liu et al., "Recurrent Neural Network for Text Classification with Multi-Task Learning". *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, 2016.

[15] Wang H et al., "Online Reliability Prediction via Long Short Term Memory for Service-Oriented System". *IEEE International Conference on Web Services (ICWS)*, IEEE, Honolulu, Hawaii, USA, pp.81-88, 2017.

[16] Li S et al., "A Method of Emotional Analysis of Movie Based on Convolution Neural Network and Bi-directional LSTM RNN". *IEEE Second International Conference on Data Science in Cyberspace*, IEEE, Shenzhen, China, 2017, pp.156-161.

[17] Jason Brownlee, "Text Generation With LSTM Recurrent Neural Networks in Python with Keras". Retrieved from - *https://machinelearningmastery.com/text-generation-lstm-recurrent-neural-networks-python-keras/*, on August 2020.

[18] S. Hochreiter et al., "Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies". *A Field Guide to Dynamical Recurrent Neural Networks*, IEEE Press, 2001.

[19] Richard Socher et al., "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks". *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Beijing, China, July 26-31, 2015, pp.1556–1566,.

[20] Hongfan ye et al., "Web Services Classification based on Wide & Bi-LSTM Model", *DOI 10.1109/ACCESS.2019.2907546*, IEEE Access, 2019.

[21] Cheng H T et al., "Wide & deep learning for recommender systems". *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, ACM, Boston, MA, USA, 2016, pp. 7-10.

[22] Shi M et al., "Functional and Contextual Attention-based LSTM for Service Recommendation in Mashup Creation". *IEEE Transactions on Parallel and Distributed Systems, Vol.* 30 No. 5, 2018, pp. 1077 - 1090.

[23] Muhammad Zulqarnain et al., "Efficient Processing of GRU Based on Word Embedding for Text Classification". *International Journal on Informatics Visualization,* JOIV, *Vol.* 3 No. 4, 2019, pp. 377-383.

[24] Peiyu Liu et al., "An Improved Approach for Text Sentiment Classification Based on a Deep Neural Network via a Sentiment Attention Mechanism". *Future internet-MDPI*, April 2019.

[25] Vapnik, V "Principles of risk minimization for learning theory", Advances *in Neural Information Processing Systems,* 1992, pp. 831–838.

[26] Kim, K.J et al., "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index". *Expert System Applications, Vol.*19, 2000, pp. 125–132.

[27] Dixon. M, "Sequence classification of the limit order book using recurrent neural networks" *Journal of Computer Science*, *Vol.* 24, 2017, pp. 277–286.

[28] Pal. S.K et al., "Genetic Algorithms for Pattern Recognition". *CRC Press: Boca Raton*, FL, USA, 1996, pp. 336.

[29] Mohammed Amine Janati Idrissi et al., "Genetic algorithm for neural network architecture optimization". *3rd International Conference on Logistics Operations Management (GOL)*, DOI: 10.1109/GOL.2016.7731699, May 2016.

[30] Yanan Sun et al., "Automatically Designing CNN Architectures Using Genetic Algorithm for Image Classification". *arXiv:1808.03818v3 [cs.NE] 27*, Mar 2020.