International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Rauch-Tung-Striebel Based Tucker Feature Selection for Educational Performance Analysis

C. Vivek[1], P. Tamilselvan[2]

Submitted: 19/01/2024 **Revised**: 28/02/2024 **Accepted**: 05/03/2024

Abstract: DM methods have newly obtained an enormous attention in education field. Through immense amount of information in educational dataset, forecasting students' performance has increased further complexity. Educational DM (EDM) is necessary manner of forecasting academic achievement, blemishing hidden patterns in educational information, improving learning as well as teaching surroundings. Existing techniques were not capable to carry out prognostic analysis through superior accuracy, as well as minimum time utilization owing to their deprived performance in eradicating noise and choosing important features for additional processing. To mention forecast problems to analysis of student performance, design a new ML method called Rauch–Tung–Striebel based Tucker Cox Regressive Feature Selection (RTS-TCRFS). Data pre-processing and feature selection are used to perform educational performance analysis through superior accuracy in minimum time which outcomes in improved forecast of student academic performance. At first, number of student data is gathered as of provided dataset. After that, Rauch–Tung–Striebel Data Pre-processing (RTSDP) process is used for carry out data pre-processing. Data pre-processing is performed to remove noise, as well as miss data values for dimensionality reduction. Now, Rauch-Tung-Striebel fixed-interval smoother create it simple to modernize precedent information through novel observations. RTSDP Smoother is employing forward pass as well as reverse recursion smoother depend on EKF. Once information has processed, Tucker Congruence Cox Regressive Feature Selection is designed to carry out second function of feature selection through enhanced accuracy. Tucker Congruence Coefficient establishes resemblance of extracted features across varied samples to select pertinent features. Therefore, RTS-TCRFS technique improves outcomes of forecast of student academic performance. Experimental assessment is performed with pre-processing time, feature selection accuracy, as well as space complexity. Examined outcomes demonstrate which performance of RTS-TCRFS technique is enhanced compared to existing methods..

Keywords: *Data Mining, Rauch-Tung-Striebel Smoother, Feature Selection, Tucker Congruence Cox Regression*

1. Introduction

DM is field of discovery pioneering as well as potentially important data from dataset. Application of DM methods in region of education has achieved enormous concentration in modern years. It plans to obtain novel trends as well as original patterns as of huge datasets through ML methods. Idea of forecast tries to find out students' performance in institutions. Lately, numerous ML techniques are designed to carry out student academic performance forecast. But, accuracy of was not enhanced.

For forecasting and observe student graduation performance, neuro-fuzzy method was designed [1]. It finds out student's graduation GPA as pointer of student achievement. However accuracy of forecast was not enhanced effectively. For carry out students' performance in every day, ANN-LSTM was designed in [2]. In spite of method enhanced accuracy of forecast, data pre-processing time was not minimized.

Relevance diversity method and adapted naive Bayes algorithm were designed in [3] for forecast. But, time complexity was not minimized. Hybrid Structural Equation Modeling and ANN (SEM-ANN) was developed in [4] for students' academic accord. SEM was used to study research hypotheses. However, additional feature of academic accord were not enclosed in forecast.

For discovering student who loafer in university classes, Two-layer ensemble ML method was designed in [5]. However, feature investigation learn was not considered. New copula-based modeling technique was developed in [6] to examine authority of constant assessment scores. However, time utilization was not minimized.

For predicting student performance in superior education, novel t-SNE was used in [7]. However, entire performance of forecast was not enhanced. For finding out student performance, learning analytics system based on transformer encoder was developed in [8]. However, accuracy was minimum.

For superior Education Students, student Satisfaction Level forecast method was designed in [9][21]. Although it selects lesser number of remarkable aspects, noise data elimination was not completed. ANN was presented [10] to discover student's final performance. However, feature selection was not carried out to reduce complexity of forecast.

To resolve the exceeding problem, RTS-TCRFS technique is designed for student performance forecast. Rauch–Tung–Striebel Data Pre-processing is used to eliminate noise information as well as handles missing values in dataset by calculating connection among data through lesser time.

Tucker Congruence Cox Regressive Feature Selection is employed to choose pertinent features through examining association among features as well as in addition eradicate irrelevant one to enhance accuracy[23].

[1]*Department of Computer Science,Karpagam Academy of Higher Education,Coimbatore-641 021.Email:Vivekcmca@gmail.com*
[2]*Department of Computer Science,Karpagam Academy of Higher Education,Coimbatore-641 021Email:.tamilselvancs@kahedu.edu.in*

Wide as well as comparative experiments were performed to estimate performance of RTS-TCRFS technique through dissimilar parameters.

Reminder of manuscript is structured to dissimilar part. Section 2 reviews literature survey in student performance forecast. Section 3 explains RTS-TCRFS technique through diagram. Section 4 presents experimental system. Section 5 provides results through varied parameters and finally, section 6 gives summary of manuscript.

2. Literature survey

For calculating danger of students, hybrid method by features extracted as of graph depiction of data through GCN was presented in [11]. However, computational complexity was not minimized. Attention-based NN were designed in [12] to establish student learning outputs. But, time taken for learning result discovery remained superior.

Hybrid SEM-ML method was developed [13] to discover student perception for medical educational uses. But, performance of perception recognition was not enhanced. New ML-based methods were presented in [14] to identify last test grades of students. Other than, forecast through important features was not concerted.

For observing performance change ability between students Bootstrap resampling was developed in [15]. However, accuracy remained unaddressed. For addressing multi-skill problems, XGBoost technique was examined in [16]. But, it failed to utilize bigger databases.

To mention this problem, end-to-end DL method was developed [17] to extort features. But, additional time was considered for feature selection. However, another DL-based approach was designed in [18][22] for attaining forecast accuracy. Yet, it failed to choose important features.

For discovery students' meetings , Graph CN was developed in [19]. However, resemblances between knowledge concepts were not calculated. For comprising forecast of human decisions, hybrid ML approach was explored in [20]. Other than, prediction performance was not improved.

3. Proposed Methodology

DM is learn to extort novel as well as probably important data from enormous amounts of information. In contemporary days, EDM has helpful device to study hidden patterns in educational data, forecast university attainment, as well as enhance learning environment. Information extracted as of educational domain presents novel chances to enhance forecast of student performance. Lately, ML has achieved important concentration in forecast process. However, research on traditional forecast techniques is not sufficient to discover finest techniques for predicting student's performance. New technique named RTS-TCRFS is developed for student academic performance forecast.

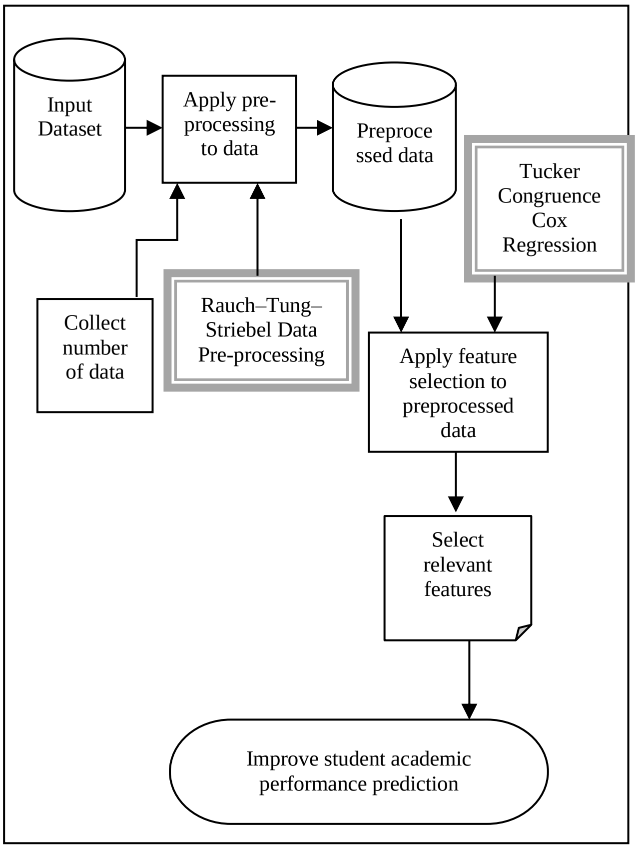

Fig. 1 Architecture of the proposed RTS-TCRFS method

Fig 1 depicts architecture of RTS-TCRFS technique to enhance academic performance forecast through superior accuracy. To begin through this, numerous information as well as their aspects are gathered as of input dataset. After that, pre-processing is used to input data by RTSDP method for eliminating noise as well as missing data values. Afterward this, TCRFS process is used on preprocessed information to select features which are further revealing for student performance forecast. This, in turn, superior student academic performance forecast is attained through minimum time.

3.1 RTSDP

Data pre-processing is initial method for accurate study of input information for additional processing. It identifies as well as exact data that is missing data. Data attained as of dataset have errors as well as noises owing to assortment of features. So, processing of such information gives wrong outcomes in forecast process. Therefore, data pre-processing is carried out to form noise-free data for simple as well as precise processing. RTSDP method is used to perform noise removal as well as handle missing values.

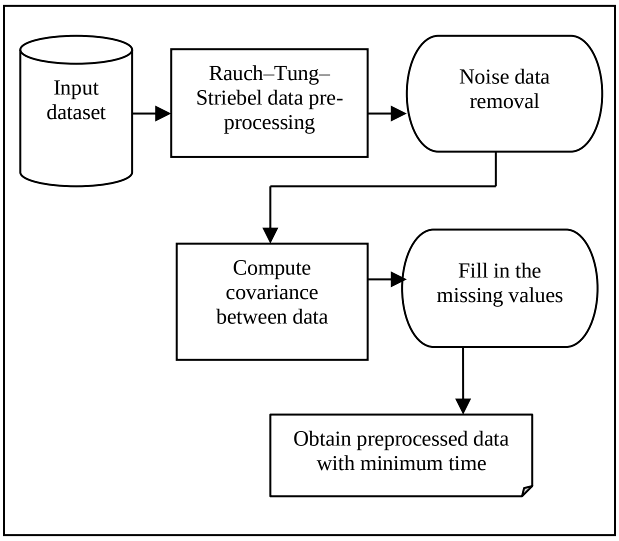

Fig. 2 Rauch–Tung–Striebel Data Pre-processing Model

Fig 2 depicts block diagram of data pre-processing by RTSDP method. Rauch–Tung–Striebels moother is non-linear filtering method which aids to remove noise from dataset. RTSDP is functioned depend on forward pass as well as reverses recursion smoother. Forward pass is functioned depend on EKF. Reverse pass calculates covariance among data. Through using EKF, noise is processed as follows.

$$A_{t+1} = S\,A_t + C_t F_t + d_t \vartheta_t \tag{1}$$

In equation (1), 'A_{t+1}' denotes updated filtering result, 'S' represents state transition function, 'A_t' indicates present state, 'C_t' denotes control input function, 'F_t' represents control input, 'd_t' is noise input data, 'ϑ_t' symbolize process noise. Noise in dataset is eradicated to obtain improved-quality data. This can complete through arranging gathered input data '$d = d_1, d_2, d_3 \ldots, d_n$' as well as their features in matrix format as follows,

$$X = \begin{bmatrix} d_{11} & d_{12} & \ldots & d_{1m} \\ d_{21} & d_{22} & \ldots & d_{2m} \\ d_{31} & d_{32} & \ldots & d_{3m} \\ d_{n1} & d_{n2} & \ldots & d_{nm} \end{bmatrix} \tag{2}$$

From equation (2), gathered data is indicated as matrix 'X'. Here, every row is equal to 'n' number of data in dataset ('n'), every column is equal to 'm' number of features ('m') in dataset. After that, denoising is computed as below.

$$G_{(d_k|d_{k-1})} = \frac{1}{var\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{d_k-\mu}{var}\right)^2\right) \tag{3}$$

In equation (3), '$G_{(d_k|d_{k-1})}$' denotes filtering output, 'd_k' represents data at present state, 'd_{k-1}' symbolize data in preceding state, 'var' specifies variance as of mean, 'μ' indicates mean.

After that, mean as well as variance value of data 'μ' is calculated as below.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} d_i \tag{4}$$

$$var(d_i) = (d_i - \mu_i)^2 \tag{5}$$

From equation (4) , (5) μ as well as variance value of data is calculated. Through this, data which diverge from mean is represented to noise data. This noise information is abandoned as of input dataset. Next, missing data values are performed through surrogating option data values. Substitute data value is recognized by calculating covariance among data and formulated as follows,

$$cov\left(d_i, d_j\right) = \frac{\sum(d_i-\mu)(d_j-\mu)}{n} \tag{6}$$

From equation (6), 'd_i' indicates value of data in i^{th} row, 'd_i' specifies value of information in 'j^{th} row, 'μ_i' points outs the mean of the data 'd_i' and 'μ_i' indicates the mean of the data 'd_i' and 'n' symbolizes a total number of the data. From equation (6), the data that is highly correlated to the neighboring data is considered to fill the missing data. It is given by,

$$O_p = max\left[cov\left(d_i, d_j\right)\right] \tag{7}$$

In the above equation (7), 'O_p' indicates a pre-processing result, '$max\left[cov\left(d_i, d_j\right)\right]$'maximum covariance between data. Accordingly, the missing values are handled to get the smooth preprocessed data. Pseudo code for Rauch–Tung–Striebel data pre-processing is provided as follows.

//Algorithm 1: Rauch–Tung–Striebel Data Pre-Processing
Input: Input dataset 'D', number of data '$d = d_1, d_2, d_3 \ldots, d_n$', number of features '$\beta_1, \beta_2, \beta_3, \beta_4 \ldots \ldots, \beta_m$'
Output: ObtainPre-processed data
Begin **1.** Collect the number of data '$d = d_1, d_2, d_3 \ldots, d_n$' and features '$\beta_1, \beta_2, \beta_3, \beta_4 \ldots \ldots, \beta_m$' from the dataset **2.** Arrange data and features into matrix 'X' as in (2) **3.** Apply data pre-processing model **4.** Measure the relationship between data 'd_k' and mean 'μ' using (3) **5.** **If** (data is closer to mean) **then** **6.** data is said to be noise-free data **7.** **Else** **8.** data is deviate from the mean **9.** data is said to be noise data **10.** remove the noise data **11.** **End if** **12.** Compute covariance between data '$cov\left(d_i, d_j\right)$' using (6) **13.** Fill the missing data by satisfying equation (7) **14.** Obtain preprocessed data **End**

Algorithm 1 shows the process of data pre-processing to get noise-free data using Rauch–Tung–Striebel data pre-processing. With the objective of eliminating noise and handling missing values with minimum time, two different functions are used. First, the noise data is discarded from the dataset by calculating the relationship between data in the current data and the mean value of the data. After that, the missing data filling is performed by computing the covariance between data and their neighboring data. As a result, the data pre-processed results are obtained to minimize the time taken in the student performance prediction process.

3.2 Tucker Congruence Cox Regressive Feature Selection

Upon completing the data pre-processing, feature selection is employed in RTS-TCRFS method for choosing pertinent features to strengthen the prediction results. It is a procedure of getting pertinent features from an enormous set of features. In this work, Tucker Congruence Cox Regressive Feature Selection (TCCRFS) is carried out to select the pertinent features with higher accuracy. Here, features that contribute enough for predicting variable or output (i.e., student performance prediction) are selected robotically. TCCRFS is designed with the advantages of speed up a data mining algorithm, increasing the data quality, improving the performance of data mining, and clarity of the mining results.

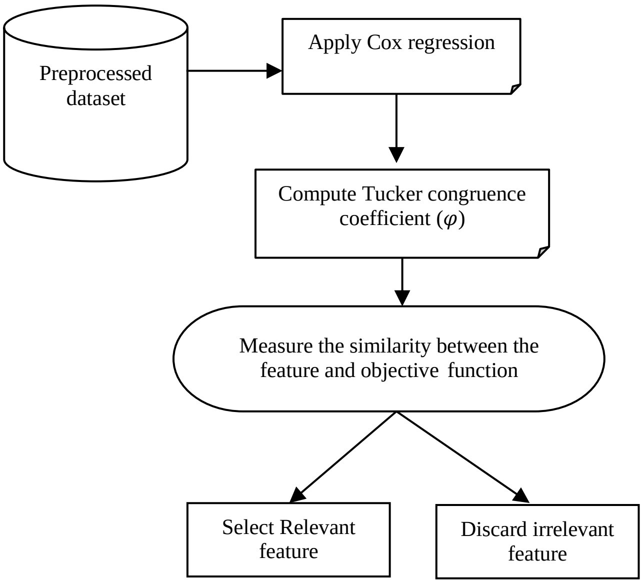

Fig. 3 Tucker Congruence Cox Regressive Feature Selection

Figure 3 illustrates the process of Tucker congruence cox regressive feature selection to minimize the computational complexity problem. The cox regression is a statistical method to compute the relationships between two variables such as dependent and one or more independent variables using a Tucker Congruence Coefficient measure. The dependent variable is an outcome whereas the independent variable is a feature in the preprocessed dataset.

Let consider, a number of features in the preprocessed dataset 'PD' as input and expressed as follows.

$$\beta_1, \beta_2, \beta_3, \beta_4 \dots \dots, \beta_m \in PD \quad (8)$$

Where 'β_m' indicates 'm' number of features in dataset 'PS '. After that, cox regression is employed to find out similarity among features with aid of tucker congruence coefficient measure and expressed as below.

$$\varphi = \frac{\sum \beta_i \gamma_j}{\sqrt{\sum \beta_i^2 \sum \gamma_j^2}} \quad (9)$$

From equation (9), 'φ' denotes the tucker congruence coefficient, and it is measured between the input feature and objective function. Here, 'β_i' denotes the input feature and 'γ_j' denotes the objective function (i.e.., student academic performance prediction). The results of the Tucker Congruence Coefficient ranged between -1 and +1.

$$Y_\beta = \begin{cases} \varphi = 0 \text{ to } +1; & R_\beta \\ \varphi = -1 \text{ to } 0; & I_\beta \end{cases} \quad (10)$$

Where, 'Y_β' denotes an output value of Tucker Congruence Coefficient, 'R_β' denotes a relevant feature and 'I_β' denotes an irrelevant feature. If the output of the tucker congruence coefficient value falls between 0 and 1, then the feature is said to be relevant for student academic performance prediction. Otherwise, the feature is claimed as irrelevant to the prediction process. The relevant features are only considered for the prediction process whereas the irrelevant features are discarded. From that, the TCCRFS decreases the time and space complexity involved in the student academic performance prediction process.

Table 1: Features selected using Tucker Congruence Cox Regression

S.No	Selected Features
1	Marital status
2	Course
3	Previous qualification
4	Previous qualification (grade)
5	Gender
6	Scholarship holder
7	Age at enrollment
8	Curricular units 1st sem (approved)
9	Curricular units 1st sem (grade)
10	Curricular units 2nd sem (approved);
11	Curricular units 2nd sem (grade);
12	Unemployment rate
13	Inflation rate
14	GDP
15	Target

Pseudo code for Tucker Congruence Cox Regressive Feature Selection is described as follows.

//Algorithm 2: Tucker Congruence Cox Regressive Feature Selection

Input: Preprocessed dataset 'PD'; number of features '$\beta_1, \beta_2, \beta_3, \beta_4 \dots \dots, \beta_m$'

Output: Select relevant features

Begin

1. **Initialize** number of input features '$\beta_1, \beta_2, \beta_3, \beta_4 \dots \dots, \beta_m$' from '$PD$'

2. **For** each input feature 'β_i'

3. **Apply** Cox correlation

4. Compute Tucker Congruence coefficient

5. **If** (φ = 0 to 1), **then**

6. Identify feature as relevant 'R_β'

7. Select the feature

8. **Else**

9. Identify feature as irrelevant 'I_β'

10. Remove the feature

11. **End If**

12. **End for**

End

As explained in above algorithm 2, the process of relevant feature selection is carried out to reduce the time and space complexity using the Tucker congruence cox regressive feature selection algorithm. The number of features from the student database is obtained initially. Then the cox regression is employed to find the similarity between the features and the objective function through the Tucker congruence coefficient in the RTS-TCRFS method. When the tucker congruence coefficient value is ranged from 0 to +1 then the features are discovered as relevant. Otherwise, the feature is said to be irrelevant to student academic performance prediction. With this, the irrelevant features are removed in the RTS-TCRFS method. As a result, the time and space complexity contributed to the prediction process is reduced in the RTS-TCRFS method than the other methods.

4. Experimental Setup

Experimental evaluation of the proposed RTS-TCRFS method and existing Neuro-fuzzy model [1] and Artificial Neural Network and Long Short-Term Memory (ANN-LSTM) [2] are implemented in Python. The results of student academic performance prediction using the proposed RTS-TCRFS method with existing methods are analyzed by using the Students Dropout and Academic Success Dataset. The dataset is collected from https://www.kaggle.com/datasets/mahwiz/students-dropout-and-academic-success-dataset. The dataset contains 4424 instances and 37 features or attributes such as Marital status, Application mode, Application order, Course, Daytime/evening attendance, Previous qualification, Previous qualification (grade), Nacionality, Mother's qualification, Father's qualification, Mother's occupation, Father's occupation, Admission grade, Displaced, Educational special needs, Gender, Scholarship holder, Age at enrollment, International, Curricular units 1st sem (credited), Inflation rate, GDP, Target and many more. From the dataset, the data is collected and processed for student academic performance prediction,

To verify the effectiveness of this research work in the prediction of student academic performance, a comparative analysis is carried out between the proposed RTS-TCRFS method, and two existing methods such as the Neuro-fuzzy model[1], and ANN-LSTM [2]. The performance evaluation of the proposed RTS-TCRFS method is obtained based on the following crucial metrics.

- Pre-processing time
- Feature selection accuracy
- Space complexity

5. Result and Discussions

The results of proposed RTS-TCRFS method are discussed in this section. The performance of the RTS-TCRFS method is compared with the existing Neuro-fuzzy model[1], and ANN-LSTM [2].The effectiveness of the RTS-TCRFS method is computed using various performance metrics and provided in the form of tables and graphs.

5.1 Performance of pre-processing time

The result of pre-processing time is investigated in this section. The data collected from the dataset is imbalanced due to the different categories of features in the dataset. Therefore, pre-processing is carried out for prediction process. During the data pre-processing, certain amount of time is consumed and it is unavoidable. This is referred to be pre-processing time. It is mathematically given by,

$$T_{pp} = \sum_{i=1}^{n} d_i * T\ (Sd) \tag{11}$$

From the above equation (11), pre-processing time 'T_{pp}' is determined based on the number of data (i.e., student data) involved in the experiments 'd_i' and the time consumed in single data. It is measured in terms of milliseconds (ms).

Table 2 Comparison of Pre-processing Time

Number of data	Pre-processing Time (ms)		
	ANN-LSTM	Neuro-fuzzy model	RTS-TCRFS
400	32	27	18
800	35	30	21
1200	38	32	24
1600	40	35	27
2000	42	37	29
2400	44	40	33
2800	47	42	36
3200	50	45	40
3600	53	48	42
4000	56	50	45

Table 2 illustrates the comparative analysis of pre-processing time with respect to the number of data taken from the input dataset. The validation of pre-processing time using the proposed RTS-TCRFS method is analyzed by comparing with conventional methods such as Neuro-fuzzy model [1], and ANN-LSTM [2]. Among the three methods, the proposed RTS-TCRFS method takes a lesser amount of time to preprocess the data. Let us consider, the number of data is '400' in the first run. The time consumed for pre-processing student data are observed to be '18 ms ' using RTS-TCRFS method whereas '27 ms' and '32 ms' time are observed using existing [1] and [2] respectively. In this way, the various performance results of pre-processing time with respect to the number of data are observed. These observed values are compared with conventional methods. The average result of pre-processing time using the proposed RTS-TCRFS method is reduced by 20% when compared to [1] and 29% when compared to [2] respectively.

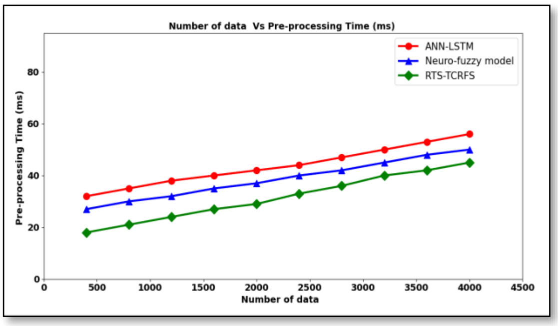

Fig. 4 Depicts pre-processing time

Figure 4 depicts pre-processing time for proposed and existing methods to predict student academic performance. As given in the abovefigure, numbers of data are taken in the horizontal axis and results of pre-processing time using RTS-TCRFS method, Neuro-fuzzy model [1], and ANN-LSTM [2] are obtained in the y-axis. Different colors of line in the graph represent the pre-processing time of proposed and existing [1] and [2] methods. On the contrary to other methods, RTS-TCRFS method uses the minimum time to preprocess the given data. The lesser time consumption during the data pre-processing is achieved by applying Rauch–Tung–Striebel Data Pre-processing process. Here, noise removal is initially performed using computing the mean value of the data. The higher difference between mean and data is identified as noise and it is discarded from the dataset. Subsequently, covariance measure (i.e., the relationship between data with their neighbor) is employed to simply handle the missing data values. This in turn, the noise and missing data values are addressed with lesser time and thereby reducing the pre-processing time in RTS-TCRFS method than the [1] and [2].

5.2 Performance of feature selection accuracy

To evaluate the predictive model, accuracy is an essential part. In this research, student academic performance prediction is assessed in terms of feature selection accuracy. It is another evaluation metric and it is computed as the ratio of the student data accurately predicted to the total count of data. The feature selection accuracy is estimated as given below.

$$Acc_{FS} = \sum_{i=1}^{n} \frac{nd_{cs}}{d_i} * 100 \tag{12}$$

From the above equation (12), 'Acc_{FS}' indicates a feature selection accuracy, 'nd_{cs}' refers a number of data where features are correctly selected, and 'd_i' 'represents the number of data in the experiments. It is measured in terms of percentage (%).

Table 3 Comparison of feature selection accuracy

Number of data	Feature selection accuracy (%)		
	ANN-LSTM	Neuro-fuzzy model	RTS-TCRFS
400	80	83	91
800	78	82	90
1200	77	81	89
1600	79	83	90
2000	78	84	91
2400	76	82	90
2800	75	81	89
3200	77	83	91
3600	76	82	92
4000	77	83	90

Table 3 describes the performance of feature selection accuracy based on the amount of data collected from the dataset. The results of feature selection accuracy of the proposed RTS-TCRFS method are compared with the existing Neuro-fuzzy model [1], and ANN-LSTM [2]. From the above-tabulated values, the feature selection accuracy of the RTS-TCRFS method is significantly improved than the existing techniques. Let's consider, 400 data in the first run. The feature selection accuracy of RTS-TCRFS method is measured as 91%, and the feature selection accuracy of the existing Neuro-fuzzy model [1], and ANN-LSTM [2] is obtained as 83% and 80% respectively. Likewise, ten different results feature selection accuracy is computed and compared. The compared average values of feature selection accuracy are improved by 10% and 17% than the existing [1] and [2] respectively.

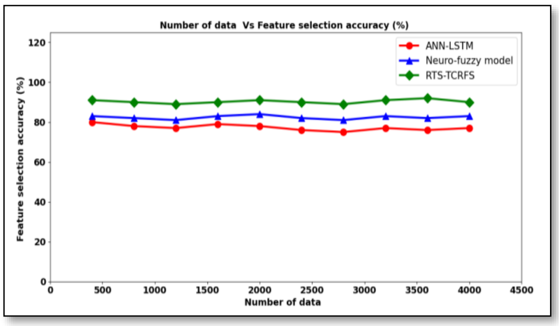

Fig. 5 Results of feature selection accuracy

Figure 5 demonstrates the results of feature selection accuracy versus a number of data for the RTS-TCRFS method, Neuro-fuzzy model [1], and ANN-LSTM [2]. The x-axis depicts the number of data (student data), whereas the y-axis represents their corresponding feature selection accuracy computed in percentage. From the graphical representation, RTS-TCRFS method exhibits higher feature selection accuracy upon comparison with [1] and [2]. The main reason behind the improvement is to apply Tucker congruence cox regressive feature selection. The feature selection method computes the relationshipsbetween two variables such as feature and objective function using the tucker congruence coefficient. With the computed coefficient results, relevant features are accurate whereas the irrelevant features are ignored in the prediction process. Accordingly, the accuracy of feature selection is improved in the RTS-TCRFS method.

5.3 Performance of f space complexity

Lastly, space complexity is computed as the amount of memory space used for performing student data prediction. The space complexity is mathematically determined as given below.

$$Sp_{com} = \sum_{i=1}^{n} d_i * memory(singledata) \qquad (13)$$

From the above equation (13), 'Sp_{com}' refers a space complexity and it is computed in megabytes (MB).

Table 4 Comparison of Space complexity

Number of data	Space complexity (MB)		
	ANN-LSTM	Neuro-fuzzy model	RTS-TCRFS
400	36	32	24
800	38	34	28
1200	42	37	30
1600	45	39	32
2000	48	42	36
2400	51	45	39
2800	54	49	41
3200	56	51	44
3600	59	53	47
4000	60	55	49

Table 4 reveals the comparative analysis of space complexity versus the number of data for three different methods. To perform the experiments, a number of data is taken as input and it is varied in the ranges of 400 to 4000. The performance of space complexity of three different methods namely the proposed RTS-TCRFS method, Neuro-fuzzy model [1], and ANN-LSTM [2] are improved with the increase in a number of data. But comparatively, the proposed RTS-TCRFS method provides lesser space complexity while predicting the student academic performance through the feature selection process. Let us consider that number of data is 400 in the first. The space complexity of the proposedRTS-TCRFS method is attained as 24 MB, and the space complexity of Neuro-fuzzy model [1], and ANN-LSTM [2]is provided as 32 MB and 36MB correspondingly. Totally ten different space complexity outcomes are determined for each method. From that, the average comparison of space complexity using RTS-TCRFS method is reduced by 16% and 25% when compared to existing [1] and [2] respectively.

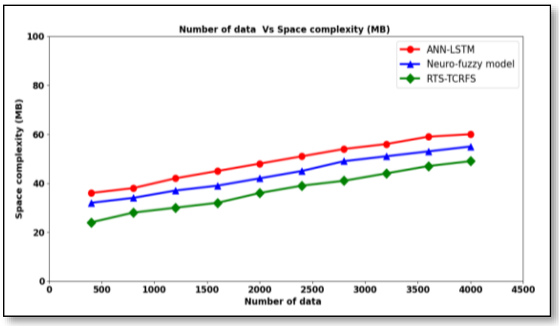

Fig. 6 Results of space complexity

Figure 6 illustrates the comparative analysis of the space complexity versus a number of data. The results of space complexity using RTS-TCRFS method are compared with the conventional Neuro-fuzzy model [1], and ANN-LSTM [2]. The above figure clearly illustrates that the space complexity of the three methods is directly proportional to the number of data. It means while an increasing number of data, memory consumed for prediction is also improved. On the contrary to existing methods, RTS-TCRFS method achieved smaller space complexity. This better performance is achieved using pre-processing and feature selection processes. Initially, pre-processing is performed by the

Rauch-Tung-Striebel model to eradicate the noisy data and thus minimize the dataset dimensionality. Then, the feature selection process is performed via Tucker Congruence Cox Regressive Feature Selection where the relevant features are only chosen to decrease the space complexity involved in the RTS-TCRFS method.

6. Conclusion

A new method called RTS-TCRFS is developed in this research work to predict student academic performance with higher accuracy. The RTS-TCRFS method incorporates two different steps such as data preprocessing and feature selection. Rauch–Tung–Striebel Data Pre-processing process is employed to eliminate the noise data and handles the missing values in the dataset. This helps to change the dataset into easily understandable and also achieve dimensionality reduction. The feature selection process is carried out with the help of Tucker Congruence Cox Regression. The association between features and the objective function is computed to choose the relevant features. The designed regression-based method not only chooses the pertinent features it also discards the irrelevant features from the dataset. With this, the feature selection accuracy is improved and thereby space used for student performance prediction is minimized. The performance of the RTS-TCRFS method is evaluated using different metrics such as pre-processing time, feature selection accuracy, and space complexity. The obtained results demonstrate that the proposed RTS-TCRFS method improved the feature selection accuracy with lesser pre-processing time and space complexity than the conventional methods.

References

[1] Riyadh Mehdi, MirnaNachouki, “A neuro-fuzzy model for predicting and analyzing student graduation performance in computing programs”, Education and Information Technologies, Springer, Volume 28, 2023, Pages 2455–2484

[2] Fatima Ahmed Al-azazi, MossaGhurab, “ANN-LSTM: A deep learning model for early student performance prediction in MOOC”, Heliyon, Elsevier, Volume 9, Issue 4, 2023, Pages 1- 16

[3] M. Shaheen , N. Naheed, A. Ahsan, “Relevance-diversity algorithm for feature selection and modified Bayes for prediction”, Alexandria Engineering Journal, Elsevier, Volume 66, 1 March 2023, Pages 329-342

[4] SoolmazAghaei, YaserShahbazi, MohammadtaghiPirbabaei, Hamed Beyti, “A hybrid SEM-neural network method for modeling the academic satisfaction factors of architecture students”, Computers and Education: Artificial Intelligence, Elsevier , Volume 4, 2023, Pages 1- 18

[5] Jovial Niyogisubizo, Lyuchao Liao, Eric Nziyumva, EvaristeMurwanashyaka, Pierre ClaverNshimyumukiza, “Predicting student’s dropout in university classes using two-layer ensemble machine learning approach: A novel stacked generalization”, Computers and Education: Artificial Intelligence, Elsevier, Volume 3, 2020, Pages 1-12

[6] Thong Nguyen-Huy, Ravinesh C. Deo, Shahjahan Khan, Aruna Devi, Adewuyi Ayodele Adeyinka, Armando A. Apan, And Zaher Mundher Yaseen, “Student Performance Predictions for Advanced Engineering Mathematics Course With New Multivariate Copula Models”, IEEE Access, Volume 10, 2022, Pages 45112-45136

[7] Essa Alhazmi and Abdullah Sheneamer, “Early Predicting of Students Performance in Higher Education”, IEEE Access, Volume 11, 2023, Pages 27579-27589

[8] Sri Suning Kusumawardani, and Syukron Abu IshaqAlfaroz, “Transformer Encoder Model for Sequential Prediction of Student Performance Based on Their Log Activities”, IEEE Access, Volume 11, 2023, Pages 18960-18971

[9] Hanan E. Abdelkader, Ahmed G. Gad, Amr A. Abohany, And Shaymaa E. Sorour, “An Efficient Data Mining Technique for Assessing Satisfaction Level With Online Learning for Higher Education Students During the COVID-19”, IEEE Access, Volume 10, 2022, Pages 6286-6303

[10] Tarik Ahajjam, Mohammed Moutaib, Haidar Aissa, Mourad Azrour, Yousef Farhaoui,and Mohammed Fattah, “Predicting Students’ Final Performance Using Artificial Neural Networks”, Big Data Mining and Analytics, IEEE, Volume 5, Issue 4, December 2022, Pages 294 – 301

[11]BalqisAlbreiki, Tetiana Habuza1 and NazarZaki, “Extracting topological features to identify at-risk students using machine learning and graph convolutional network models”, International Journal of Educational Technology in Higher Education, Springer, Volume 20, 2023, Pages 1-22

[12] Eugene Yujun Fu, Grace Ngai, Hong Va Leong, Stephen C.F. Chan & Daniel T.L. Shek, “Using attention-based neural networks for predicting student learning outcomes in service-learning”, Education and Information Technologies, Springer, 2023, Pages 1-27

[13]Amina Almarzouqi, Ahmad Aburayya, and Said A. Salloum, “Prediction of User’s Intention to Use Metaverse System in Medical Education: A Hybrid SEM-ML Learning Approach”, IEEE Access, Volume 10, 2022, Pages 43421 – 43434

[14]Mustafa Yagci, “Educational data mining: prediction of students’ academic performance using machine learning algorithms”, Yağcı Smart Learning Environments, Springer, Volume 9, Issue 11, 2022, Pages 1-19

[15]Roberto Bertolini, Stephen J. Finch, Ross H. Nehm, “Quantifying variability in predictions of student performance: Examining the impact of bootstrap resampling in data pipelines”, Computers and Education: Artificial Intelligence, Elsevier, Volume 3, 2022, Pages 1-12

[16]Wei Su, Fan Jiang, Chunyan Shi, Dongqing Wu, Lei Liu, Shihua Li, Yongna Yuan & Juntai Shi, “An XGBoost-Based Knowledge Tracing Model”, International Journal of Computational Intelligence Systems, Springer, Volume 16, Issue: 13, 2023, Pages 1-9

[17] Xiaoyong Li, Yong Zhang, Huimin Cheng, Mengran Li, Baocai Yin, “Student achievement prediction using deep neural network from multi-source campus data”, Complex & Intelligent Systems, Springer, Volume 8, 2022, Pages 5143–5156.

[18] Mohammed Jarbou, DaehanWon, JenniferGillis-Mattson & Raymond Romanczyk, “Deep learning-based school attendance prediction for autistic students”,Scientific Reports, Springer, Volume 12, Issue 1431, 2022, Pages 1-11

[19] Ahmed A. Mubarak, Han Cao, Ibrahim M. Hezam, Fei Hao, “Modeling students’ performance using graph convolutional networks”, Complex & Intelligent Systems, Springer, Volume 8, 2022, Pages 2183–2201

[20] Dursun Delen, Behrooz Davazdahemami, Elham RasouliDezfouli, “Predicting and Mitigating Freshmen Student Attrition: A Local-Explainable Machine Learning Framework”, Information Systems Frontiers, Springer, 2023, Pages 1-22

[21] Sultanuddin, S. J., Devulapalli Sudhee, Priyanka Prakash Satve, M. Sumithra, K. B. Sathyanarayana, R. Krishna Kumari, Jonnadula Narasimharao, R. Reddy, and R. Rajkumar. "Cognitive computing and 3D facial tracking method to explore the ethical implication associated with the detection of fraudulent system in online examination." *Journal of Intelligent & Fuzzy Systems* Preprint (2023): 1-15.

[22] Lakshmi Narayanan, S., Shunmugavel, S., Prasanth, R., Satheesh Kumar, M., Srujan Raju, K., Suthendran, K. (2023). Design and Implementation of Cyber Threat Intelligence Data Mining Model. In: Satapathy, S.C., Lin, J.CW., Wee, L.K., Bhateja, V., Rajesh, T.M. (eds) Computer Communication, Networking and IoT. Lecture Notes in Networks and Systems, vol 459. Springer, Singapore. https://doi.org/10.1007/978-981-19-1976-3_22

[23] Joshi, A., Choudhury, T., Sai Sabitha, A., Srujan Raju, K. (2020). Data Mining in Healthcare and Predicting Obesity. In: Raju, K., Govardhan, A., Rani, B., Sridevi, R., Murty, M. (eds) Proceedings of the Third International Conference on Computational Intelligence and Informatics . Advances in Intelligent Systems and Computing, vol 1090. Springer, Singapore. https://doi.org/10.1007/978-981-15-1480-7_82