Using Factor Analysis to Increase Accuracy of Neural Networks for Predicting Student Test Scores

Michael Scott Brown
Information Systems Department
University of Maryland
Baltimore County
Baltimore, MD, USA
michaelb@umbc.edu

Maliha Momtaz
Computer Science and IT
Department
Hood College
Frederick, MD, USA
malihachaity1089@gmail.com

Aaron Mai
Computer Science and
Engineering Department
Texas A&M
College Station
College Station, TX, USA
aaronjmai@tamu.edu

Dhruvil Modi
Department of Computer Science
and Electrical Engineering
University of Maryland
Baltimore County
Baltimore, MD, USA
dhruvim1@umbc.edu

Abstract—**This work is a full research-to-practice paper that describes a predictive method to improve the prediction of student test scores. Predicting student test scores is difficult. However, doing so can improve education greatly by improving advising, scheduling, tutoring assignment and other educational processes. This research extends previous research by using a domain space reduction technique to improve accuracy. Factor Analysis is used to reduce the number of domain attributes for improving the accuracy of a Neural Network to predict student test scores.**

In this research datasets for Mathematics and Language of high school student test scores were used. Test scores were predicted using a Neural Network computing the Mean Absolute Error as a measurement of accuracy. The datasets have 30 domain attributes each. Factor Analysis was used to reduce the domain size from between 1 to 29, each time using it to train the Neural Network. Because the Mean Absolute Error may vary depending upon which records in the dataset are used for training versus testing, 50 trials of each dataset size were executed producing an Average Mean Absolute Error for each domain size. A statistical test was used to show statistical significance between the Neural Network without Factor Analysis and the Neural Network with varying domain sizes using Factor Analysis.

Results were very promising and correspond to previous research that used Principal Component Analysis. Numerous domain sizes had significantly better Average Mean Absolute Errors than the accuracy of the Neural Network without Factor Analysis.

This research shows that reducing the domain size using Factor Analysis can greatly improve the accuracy of Neural Networks when predicting student test scores. The best improvements occurred when domain sizes were very small ranging from 2 to 6. Domain reduction techniques, such as Factor Analysis, have been shown to improve predictive models for student test score prediction. Future research should consider other domain reduction techniques, like T-SNE, in combination with other predictive models, like Support Vector Machines.

Keywords—Factor Analysis, High School, Neural Networks, Predicting Student Test Scores, Domain Space Reduction, Domain Reduction, Predictive Models

I. INTRODUCTION

A. Background

Predicting students' test scores is highly beneficial for helping students enhance their performance and supporting educational institutions in upgrading their educational systems. Making predictions using machine learning (ML) and data science techniques enables educators and policymakers to make informed decisions. Large and high-quality datasets are necessary to train such models, and fortunately, many public datasets are available for educational research. ML, deep learning (DL), and data science models have the capability to identify relevant features and transform them appropriately through preprocessing.

However, choosing the right model is crucial, and the decision depends on the characteristics of the dataset. Educational systems are dynamic and subject to changes in curriculum, teaching methods, and assessment techniques, making it challenging to develop models that remain relevant over time. ML models exhibit the ability to adapt to new data, allowing the system to evolve and improve as more information becomes available. Predicting students' test scores holds value and shows promise for improving educational outcomes.

This paper proposes a new method to predict student test scores. Factor Analysis [1] is used to reduce the domain data for a Neural Network.

B. Problem Statement

Predicting students' test scores is challenging for various reasons [2]. It is not understood what factors contribute to student test scores [3]. Most research uses an exhaustive list of available data, which typically includes data related to prior academic performance, home life, and other activities. Many predictive models have been attempted producing various results [4], but there are still improvements needed to models to achieve even better results.

C. Importance

Predicting students' test scores has numerous benefits. By forecasting test scores, educators can identify students who may be facing academic challenges early on, enabling timely intervention and the provision of necessary support to prevent potential setbacks [5]. This early identification allows for the implementation of advanced teaching methods, resources, and assignments, creating a more effective and engaging learning experience. Educational institutions can pinpoint areas that require additional instructional support, helping them refine their curriculum and teaching approaches for the betterment of diverse student populations [6-7]. Predicting student test scores has many advantages to students and educational institutions.

D. Approach

This research focuses on improving the accuracy of students' test score predictions by pre-processing data using domain space reduction techniques prior to training the model. Through new and prior research, domain space reduction techniques like Principal Component Analysis and Factor Analysis are used with Neural Networks and Regression models. This technique proves particularly beneficial when dealing with high-dimensional data, significantly enhancing the computational efficiency of predictive models and providing faster training times.

This research will compare prior and new results related to improved accuracy of predictions using domain space reduction. Previous research used Principal Component Analysis with Neural Network and Regression models. This research contributes results from using Factor Analysis with Neural Networks.

II. Literature Review

A. Predictive Models

Predictive models are algorithms that process a multitude of variables to predict future events or outcomes. When creating predictive models, there are many different approaches that can be utilized, all of which collect and process original data to output some potential future event using models and algorithms. Listed below are some of the most popular approaches to constructing a predictive model along with how it was utilized to predict student test scores.

Over 150 research papers describe models to predict student test scores using the [8] dataset. The most used predictive models were Neural Networks, Bayesian Models, Support Vector Machines, K-Nearest Neighbor and Regression. Using Neural Networks is the most common approach. However, all of the models where used in multiple research.

1) Regression

Regression is a statistical measure and relationship between multiple variables. It is a series of models that attempt to fit the relationship found between variables in an equation. Regression places all the data on a model and tries to find the line of best fit that can be thought of as an average of all data points. There are various types of regression like Linear, Polynomial, and Logistic Regression that aim to produce constants that minimize error in the data.

The most relevant regression model to predict student grades is Logistic Regression. Accounting for factors like school, age, family size, internet access, and other attributes, [9] found Logistic Regression to be one of the most effective algorithms in predicting students' grades. Other research that uses regression to predict student test scores are [10-11].

2) Neural Networks

A Neural Network is a deep learning model directly inspired by the human brain. Humans have hundreds of billions of neurons in their brain, all interconnected in a vast network, capable of receiving and responding to stimuli. Neural Networks are similarly constructed with artificial neurons, acting as the counterpart for biological neurons and the most elementary unit in the model. These artificial neurons are assigned with coefficients, or weights, and interwoven into a network to form a Neural Network, complete with connection weights to each other. Some input is given to a Neural Network, which processes the input by multiplying it with corresponding weights and pushing it through various other layers of neurons before outputting a result. However, while a Neural Network is capable of processing information like a human brain, it first needs to determine its weight values. Base training datasets are first fed through the Neural Network to calibrate the weight values. When trained appropriately, a Neural Network can receive and respond to stimuli, mimicking human neurons.

[5] utilizes a standard Neural Network trained on a student database containing information on test scores and corresponding attributes of the test taker including age, gender, family size, parental education and occupation, educational support, health, and many more attributes. The experiment would train the Neural Network on a random subset of the public database and test it on a different random subset, repeating the process many times to reduce error. The results are then averaged and compared to determine potential future test scores given a list of parameters. Neural Networks have been one of the most popular predictive models to use for predicting student test scores. Some of the earliest work on predicting student test scores used Neural Networks [12]. Additional research using Neural Networks can be found at [10].

3) Support Vector Machines

Support Vector Machines are models that find hyperplanes that separate data points into different categories. A Support Vector Machine can be thought of as a massive Venn diagram, separating items based on their similarities and differences with each other. It creates groups, or classes, based on an aggregation of its members' properties, which can be used to find ranges of combinations of variables that yield optimal results [13].

The results of a Support Vector Machine can be used to determine the best combinations of variables to predict future events like test scores. When given parameters like age, grade, and other factors, the Support Vector Machine can find the best hyperplane that separates the data into groups to determine what has the most significant impact on education. Research using Support Vector Machines to predict student test scores includes [13-14].

4) K-Nearest Neighbor

The K-Nearest Neighbor algorithm is a prediction algorithm that takes training data and makes predictions based on groups of data points. It takes the training data and a data point in a different data set and measures the distances between each. Distance between data points can be measured multiple ways including Euclidean distance and Manhattan distance. After the distances have been computed for all data points, the K-Nearest Neighbor algorithm finds clusters of data records and finds the majority class, or the K-Nearest Neighbor predicted class. When K-Nearest Neighbor is used to predict values, like test scores, it averages the range values found in the cluster. Because of its simplicity, K-Nearest Neighbor is easy to implement.

K-Nearest Neighbor is a simple algorithm, but often leads to accurate results, which is why it has been used to predict student test scores. [15] used a variation of the algorithm called FKNN to predict student test scores. This research showed that FKNN was more accurate than a traditional K-Nearest Neighbor. [16] used K-Nearest Neighbor to classify scores into groups which corresponded to ranges of test scores. This approach was interesting because it turned the problem into a classification problem by classifying all grades into four groups.

5) Bayesian Algorithms

Bayesian Algorithms are probabilistic models that utilize Bayes Theorem to predict future events. Bayesian Algorithms calculate probability based on factors with contributions independent of each other, which makes the algorithm highly scalable, able to predict with many different attributes. It has been used to predict student test scores in research including [6, 17].

6) Random Forest

Random Forest algorithm is a machine learning algorithm that utilizes a multitude of decision trees, a hierarchal classification and regression algorithm representing all possible nodes from a series of events analogous to a flow chart, constructed during training to determine the most important features of a dataset and predict future outcomes based on them. The algorithm typically utilizes a voting technique to select a classification and a random set of data to determine the regression and find the best answer. The main benefits of this approach are ease of visualization through being sorted into "trees" and resistance to overfitting due to the application of randomness. [18] found that the utilization of the Random Forest algorithm was very accurate for student performance when combined with fuzzy logic while [Leena 2020] found that Random Forest gave the most accurate results with a smaller dataset.

In conclusion, all the above-mentioned methods have been proven to be viable methods to predict student scores and performance. However, some methods like Random Forest appear to struggle with datasets with high dimensionality. As such, methods to reduce the dimensionality of datasets prior to processing could provide better accuracy in results.

B. Domain Space Reduction

Domain Space Reduction are any techniques that reduce the number of domain values in a dataset. There are a number of reasons that this might be desirable. Reducing domains down to 3 dimensions allows it to be graphed, which provides benefits of visualization [20]. Smaller domain sizes reduce the time to train models, so there is a performance benefit [21]. Some research suggests that reducing domain spaces can enhance predictive model accuracy [22], while others indicate its effectiveness in improving student test score predictions [5, 23].

1) Principal Component Analysis

When dealing with data, there are many attributes that must be tracked and recorded to produce reliable results. As the complexity of any given experiment or situation increases, the amount of data recorded goes up as well. The result is a large data set that can be very hard and time consuming to parse through. To combat this, many different methods of reducing the dimensionality, or the number of features, of a given data set have been developed. One of the most popular methods of dimension reduction is the Principal Component Analysis, which is an algorithm that creates a smaller list of variables in the domain to test. The algorithm works by making new variables, Principal Components, combine correlated attributes together so that the resulting variables are uncorrelated and affect the result independent of each other while retaining as much information as possible [24]. This makes it possible to process the original data with fewer variables, thus increasing performance by decreasing the training time as proven in [25]. Interestingly, Principal Component Analysis manages to do this without much or any loss of information. Indeed, research by [25, 5] suggests that applying Principal Component Analysis to a dataset for Neural Networks or related models can enhance model accuracy by eliminating the need to account for potentially correlated variables. Because of this, Principal Component Analysis is often applied before processing data into a predictive model as a dimensionality reduction tool.

2) Factor Analysis

Factor Analysis is a statistical technique used to describe variability among found, correlated variables in phrases of a potentially lower range of unobserved variables known as elements. The essence of Factor Analysis lies in its capacity to perceive underlying relationships within a dataset with the aid of reducing the dimensionality of data at the same time as retaining as plenty of the authentic data as feasible. This method is particularly precious in fields such as psychology, schooling, and social sciences, where researchers are looking to discover latent constructs that are not at once observable [26].

In practice, Factor Analysis starts with the correlation matrix of the variables concerned, analyzing the diploma of correlation among every pair. The method assumes that any observed variable can be linearly related to a wide variety of unobservable latent factors plus a unique component that captures the mistake or the specific variance of the variable. The aim is to determine the least range of factors which could account for the most variance located in the dataset [1].

There are two fundamental forms of factor analysis: Exploratory Factor Analysis and Confirmatory Factor Analysis. Exploratory Factor Analysis is used when researchers do not have a predetermined notion about the structure or number of factors in their data. It permits the exploration of the feasible underlying structure. Conversely, Confirmatory Factor Analysis is implemented when researchers want to check a specific speculation about the aspect shape they consider exists inside their information, making it a theory-pushed approach.

The technique of determining the quantity of factors in Exploratory Factor Analysis involves criteria which includes the Kaiser criterion, scree plot analysis, and parallel analysis. Once factors are extracted, a rotation, such as Varimax or Oblimin, can be carried out to make the shape more interpretable via maximizing excessive loadings and minimizing low loadings on each element.

Factor Analysis is an effective device for facts discount and interpretation, enabling researchers to simplify complicated records sets into extra practicable, interpretable constructs. This method no longer best allows a deeper information of the records via revealing hidden dimensions but additionally complements the efficiency of next analyses by lowering the complexity of the facts [27].

Building upon the foundational understanding of Factor Analysis, it is essential to appreciate its practical applications and challenges. In the realm of research, Factor Analysis is often employed to validate the construct validity of tests and scales. For instance, in psychology, Factor Analysis can be used to confirm whether a set of psychological traits can be grouped into broader, more general factors, thereby aiding in the development of new psychological theories and assessments. In the field of marketing, businesses utilize Factor Analysis to identify underlying customer attitudes and preferences, guiding product development and marketing strategies. However, Factor Analysis is not without its challenges. The interpretation of factors requires subjective judgment, and the rotation methods can yield different results, emphasizing the necessity for researchers to carefully consider the theoretical framework guiding their analysis. Moreover, the success of Factor Analysis hinges on the adequacy of data; large sample sizes and the assumption of normality often underpin the robustness of the analysis. Therefore, while Factor Analysis is a powerful tool for uncovering hidden structures within complex datasets, its application demands a careful and thoughtful approach, underpinned by a solid theoretical understanding and a critical evaluation of the methodological choices made during the analysis process [28].

Factor analysis is very useful in many areas. It helps to condense large number of features into a smaller number of meaningful factors, thus simplifying complex data. Focusing on a few factors which capture most of the information helps in data reduction [1]. Patterns and relationships might be noticed by exploring underlying factors, which may not be obvious by observing individual variables. Once the underlying factors are understood, they can be used to predict future outcomes or make better decisions based on the discovered relationships. Factor analysis is a very popular tool in the fields of psychology, sociology, marketing, economics, and biology [23]. It is very useful for understanding the hidden structure when lots of data is available. Its application demands a careful and thoughtful approach, underpinned by a solid theoretical understanding and a critical evaluation of the methodological choices made during the analysis process.

III. METHODOLOGY

A. *Research Method Employed*

This research used an experimental research approach and will present and compare previously published results with new results. A publicly available dataset consisting of two files of student test scores in Mathematics and Language are used to predict students test scores. A Neural Network is trained to predict the test scores. Results of this are compared to results of a Neural Network but using data that was pre-processed using Factor Analysis. The Factor Analysis is used to reduce size of the domains for the datasets.

Because the data records are randomly divided into training and test groups, results will differ based upon which records are in each group. To mitigate this issue each test was run 50 times producing results. The results were averaged producing an Average Mean Absolute Error (AMAE). This metric is used in other research like [10]. The T-Test was used to compare the AMAE between the Neural Network without Factor Analysis and each of the Neural Network results with Factor Analysis of different domain sizes.

B. *Specific Procedures Employed*

The dataset format is shown in Table 1. It contains 30 domain attributes. The regular Neural Network will use all 30 attributes. The AMAE of 50 trials using the regular Neural Network will be used for comparison with results that include reduced domain sizes. Twenty-nine other sets of 50 trials were generated for each domain size 1 through 29, where Factor Analysis was used for the reduction.

Factor Analysis was used to reduce the domain space. This is a five-step algorithm shown below.

1.) Initialization: Begin with the observed data matrix X of dimensions n×m, where n is the number of observations, and m is the number of variables.

2.) Correlation Matrix: Compute the correlation matrix of X to understand the relationships between variables.

3.) Factor Extraction: Apply the method Singular Value Decomposition (SVD) to decompose the correlation matrix. SVD is a mathematical technique that breaks down the original data matrix into three components: U, Σ and VT, where Σ contains the singular values that represent the strength of each factor.

4.) Determine Number of Factors: Choose the number of factors (k) based on criteria like the Kaiser criterion (eigenvalues > 1), scree plot, or parallel analysis. This step reduces the dimensionality of the dataset.

5.) Reconstruction: The original variables can now be approximated using a smaller number of factors. Each original variable is expressed as a linear combination of the k factors plus an error term, effectively reducing the domain size.

The dataset contains three grades for each student corresponding to the three terms used in Portuguese schools. Scores range from the lowest grade of 0 to the highest grade of 20. This research attempts only to predict the grade for the first term. Predicting student test scores before the class begin can help advise students on which classes to take and can be used to provide support services.

C. *Computational Complexity*

Since both Factor Analysis and neural networks are used for predicting test scores, the computational complexity of the algorithm is the addition of the complexity of factor analysis and neural networks. The complexity of Factor Analysis is typically $O(p^3)$, where p is the number of features in the dataset. The complexity of Neural Networks is $O(e*I*n^2*m)$, where e is the number of epochs, I is the number of layers, n is the number of neurons per layer and m is the number of rows in the dataset.

So, the total complexity of the algorithm is shown in Equation 1.

$$O(Total) = O(p^3) + O(e * I * n^2 * m) \quad (1)$$

Decreasing the number of features does not change the complexities of Factor Analysis and neural networks.

D. Resources

A dataset collection obtained from the University of California Irvine dataset repository was used. The dataset was created by [8] using test scores in both Mathematics and Language in Portugal. The domain values are shown in Table 1 as the attributes. All of these values for a student are known prior to the start of the class.

TABLE I. ATTRIBUTES FOR THE DATASET

Attribute #	Domain Values of Dataset	
	Attribute Name	***Description***
1	School	School student is in. 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira
2	Gender	'F' - female or 'M' – male
3	Age	Student's age (numeric: from 15 to 22)
4	Address Type	Student's home address type - 'U' - urban or 'R' – rural
5	Family Size	Family size - 'LE3' - less or equal to 3 or 'GT3' - greater than 3
6	Parent Cohabitation	Parent's cohabitation status - 'T' - living together or 'A' - apart
7	Mother's Education	Mother's Education - 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education
8	Father's Education	Father's Education - 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education
9	Mother's Job	Mother's Job - 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other'
10	Father's Job	Father's Job - 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other'
11	Reason School Selection	Reason to choose this school - close to 'home', school 'reputation', 'course' preference or 'other'
12	Guardian	Student's guardian -'mother', 'father' or 'other'
13	Time Travel to School	Home to school travel time - 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour
14	Weekly Study Time	Weekly study time - 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours
15	Past Class Failures	Number of past class failures - n if 1<=n<=3, else 4
16	Extra Educational Support	Extra educational support - yes or no
17	Family Education Support	Family education support - yes or no
18	Has Paid for Extra Classes	Extra paid classes within the course subject (Math or Portuguese) - yes or no
19	Extra-curricular Activities	Extra-curricular activities - yes or no
20	Attended Nursery	Attended nursery school - yes or no
21	Wants Higher Education	Wants to take higher education - yes or no
22	Internet at Home	Internet access at home - yes or no
23	Romantic Relationship	With a romantic relationship - yes or no
24	Quality of Family Relationship	Quality of family relationships - from 1 - very bad to 5 - excellent
25	Amount of Freetime	Free time after school - from 1 - very low to 5 - very high
26	Amount of Time with Friends	Going out with friends - from 1 - very low to 5 - very high
27	Weekday Alcohol Consumption	Workday alcohol consumption - from 1 - very low to 5 - very high
28	Weekend Alcohol Consumption	Weekend alcohol consumption - from 1 - very low to 5 - very high
29	Health	Current health status - from 1 - very bad to 5 - very good
30	Absences	Number of school absences - from 0 to 93)

IV. RESULTS

This results section includes previously published results along with newly obtained results. Two prior published works of research used the same methodology. [5] used Principal Component Analysis as the domain reduction technique with a Neural Network as the predictive model. [21] also used Principal Component Analysis as the domain reduction technique but used Regression as the predictive model. New findings from this research are included for Factor Analysis and Neural Networks.

A. Principal Component Analysis and Neural Networks

The Average Mean Absolute Error of using the Neural Network with all 30 student attributes was 3.30 for Mathematics. This is the average of the 50 trials. Table 2 shows the results for 50 trials for each number of attributes by reducing the domain from size of 29 through size of 1.

The Average Mean Absolute Error of using the Neural Network with all 30 student attributes was 2.50 for language. This is the average of the 50 trials. Table 2 shows the results for 50 trials for each number of attributes.

B. Principal Component Analysis and Regression

The Average Mean Absolute Error of using the Regression with all 30 student attributes was 4.74 for Mathematics. This is the average of the 50 trials. Table 3 shows the results for 50 trials for each number of attributes.

The Average Mean Absolute Error of using the Regression with all 30 student attributes was 5.46 for Language. This is the average of the 50 trials. Table 3 shows the results for 50 trials for each number of attributes.

TABLE II. THIS TABLE SHOWS THE RESULTS FOR 50 TRIALS USING PRINCIPAL COMPONENT ANALYSIS WITH NEURAL NETWORKS PREDICTING MATHEMATICS AND LANGUAGE TEST SCORES. [5]

Number of Attributes Reduced To	Average Mean Absolute Error of 50 Trials for Principal Component Analysis and Neural Networks			
	Mathematics		*Language*	
	Average Mean Absolute Error	*P-Val*	*Average Mean Absolute Error*	*P-Val*
1	2.81	< .00001	2.19	< .00001
2	2.84	< .00001	2.15	< .00001
3	3.63	< .00001	2.12	< .00001
4	3.98	< .00001	2.24	< .00001
5	4.06	< .00001	2.39	< .00001
6	3.79	< .00001	2.62	.000015
7	3.67	< .00001	2.69	.000144
8	3.59	< .00001	2.76	< .00001
9	3.51	< .00001	2.72	< .00001
10	3.56	< .00001	2.64	< .00001
11	3.57	< .00001	2.60	< .00001
12	3.57	< .00001	2.58	.000131
13	3.58	< .00001	2.59	.001118
14	3.54	< .00001	2.60	.000823
15	3.55	< .00001	2.61	.000183
16	3.53	< .00001	2.61	.00003
17	3.47	< .00001	2.63	.000025
18	3.43	.000161	2.62	< .00001
19	3.39	.003196	2.62	< .00001
20	3.39	.033392	2.60	< .00001
21	3.36	.037639	2.62	.000078
22	3.34	.135029	2.61	< .00001
23	3.33	.302241	2.59	.000019
24	3.33	.451706	2.58	.000152
25	3.30	.431243	2.56	.001866
26	3.30	.956205	2.53	.026269
27	3.25	.985397	2.51	.191574
28	3.26	.288341	2.49	.623159
29	3.24	.440608	2.47	.463359

TABLE III. THIS TABLE SHOWS THE RESULTS FOR 50 TRIALS USING PRINCIPAL COMPONENT ANALYSIS WITH REGRESSION PREDICTING MATHEMATICS AND LANGUAGE TEST SCORES. [23]

Number of Attributes Reduced To	Average Mean Absolute Error of 50 Trials for Principal Component Analysis and Regression			
	Mathematics		*Language*	
	Average Mean Absolute Error	*P-Val*	*Average Mean Absolute Error*	*P-Val*
1	2.79	>0.00001	2.79	>0.00001
2	2.76	>0.00001	2.76	>0.00001
3	2.74	>0.00001	2.74	>0.00001
4	2.77	>0.00001	2.77	>0.00001
5	2.82	>0.00001	2.82	>0.00001
6	2.91	>0.00001	2.91	>0.00001
7	2.93	>0.00001	2.93	>0.00001
8	3.04	>0.00001	3.04	>0.00001
9	3.14	>0.00001	3.14	>0.00001
10	3.27	>0.00001	3.27	>0.00001
11	3.41	>0.00001	3.41	>0.00001
12	3.56	>0.00001	3.56	>0.00001
13	3.80	>0.00001	3.80	>0.00001
14	4.14	>0.00001	4.14	>0.00001
15	4.82	0.39648	4.82	0.39648
16	5.90	>0.00001	5.90	>0.00001
17	8.83	>0.00001	8.83	>0.00001
18	1.86	>0.00001	1.86	>0.00001
19	1.23	>0.00001	1.23	>0.00001
20	7.97	>0.00001	7.97	>0.00001
21	6.57	>0.00001	6.57	>0.00001
22	5.82	>0.00001	5.82	>0.00001
23	5.36	>0.00001	5.36	>0.00001
24	5.05	>0.00001	5.05	>0.00001
25	4.84	0.02209	4.84	0.02209
26	4.68	0.10677	4.68	0.10677
27	4.57	0.00006	4.57	0.00006
28	4.50	>0.00001	4.50	>0.00001
29	4.46	>0.00001	4.46	>0.00001

C. Factor Analysis and Neural Networks

The Average Mean Absolute Error of using the Neural Network with all 30 student attributes was 3.26 for mathematics. This is similar to the results from [5], which found an average of 3.30. This is the average of the 50 trials. Table 4 shows the results for 50 trials for each number of attributes.

For language the Average Mean Absolute Error for the Neural Network without Factor Analysis was 2.50. This replicates the results from [5]. Table 4 shows the Average Mean Absolute Error of the 50 trials for each domain space size.

TABLE IV. THIS TABLE SHOWS THE RESULTS FOR 50 TRIALS USING FACTOR ANALYSIS WITH NEURAL NETWORK FOR PREDICTING MATHEMATICS AND LANGUAGE TEST SCORES.

Number of Attributes Reduced To	**Average Mean Absolute Error of 50 Trials for Factor Analysis and Neural Networks**			
	Mathematics		***Language***	
	Average Mean Absolute Error	***P-Val***	***Average Mean Absolute Error***	***P-Val***
1	2.81	>0.00001	2.14	>0.00001
2	2.81	>0.00001	2.07	>0.00001
3	3.74	>0.00001	2.05	>0.00001
4	4.29	>0.00001	2.04	>0.00001
5	4.13	>0.00001	2.14	>0.00001
6	3.70	>0.00001	2.39	0.00005
7	3.56	>0.00001	2.59	0.00539
8	3.37	0.01041	2.60	0.00015
9	3.31	0.36420	2.54	0.14556
10	3.29	0.56755	2.48	0.27091
11	3.28	0.75534	2.48	0.27500
12	3.29	0.53714	2.43	0.00110
13	3.33	0.10972	2.46	0.03444
14	3.35	0.04661	2.47	0.08992
15	3.33	0.07485	2.47	0.16265
16	3.34	0.09040	2.48	0.28757
17	3.33	0.15690	2.48	0.25297
18	3.29	0.67168	2.52	0.65087
19	3.23	0.32993	2.53	0.31704
20	3.18	0.01408	2.54	0.20555
21	3.12	0.00010	2.52	0.52627
22	3.08	>0.00000	2.51	0.81607
23	3.03	>0.00000	2.50	0.73380
24	3.01	>0.00000	2.48	0.29593
25	2.99	>0.00000	2.46	0.06369
26	2.98	>0.00000	2.44	0.00643
27	2.96	>0.00000	2.42	0.00015
28	2.97	>0.00000	2.42	0.00008
29	3.29	0.59188	2.46	0.02643

V. CONCLUSION

This research leverages previously published results for increasing the accuracy of predictive models using domain space reduction while producing new results. Factor Analysis is used to reduce the size of the domain to attempt to increase the accuracy of a Neural Network. Because results may differ based upon the split between training and test data, 50 trials of each domain size were performed. Results were compared to 50 trials of the Neural Network without Factor Analysis. A statistical test was performed showing the results to be statistically significant.

All of the results show that domain space reduction can increase the accuracy of predictive models for predicting student test scores. But not all domain sizes produce more accurate results. Some, in fact, produce less accurate results. However, all produce superior results when the domain space size gets very low, 2 through 4. This observation seems to be true regardless of the predictive model and regardless of the domain reduction technique. New results also support previous conclusions that Language scores can be predicted more accurately than Mathematics scores. It is unclear why this is thus makes it an area of future research.

Future research should investigate using domain space reduction with other models and using other domain space reduction techniques. To this date Neural Networks and Regression have been investigated. Support Vector Machines, Bayesian Models and K-Nearest Neighbor have not been investigated. Likewise, there are other domain space reduction techniques that could be employed. Isomap [29] and T-NSE [30] are two such techniques.

This research shows the promise of using domain space reduction to improve the prediction of student test scores. With this understanding improvements to educational systems can be made. Students can be placed into appropriate classes and at-risk students can be provided support services to be more successful.

REFERENCES

[1] A. B. Costello, and J. W. Osborne, “Best practices in exploratory component analysis: Four guidelines for getting the maximum out of your analysis”. *Practical Assessment, Research, and Evaluation*, 10(7), 2005, pp. 1-9.

[2] K. Yan, "Student performance prediction using XGBoost method from a macro perspective." *2021 2nd International Conference on Computing and Data Science (CDS)*. IEEE, 2021.

[3] P. Crowther, and S. Briant, "Predicting academic success: A longitudinal study of university design students." *International Journal of Art & Design Education* 40.1, 2021, pp. 20-34.

[4] H. A. Mengash, "Using data mining techniques to predict student performance to support decision making in university admission systems." *Ieee Access* 8, 2020, pp. 55462-55470.

[5] M. S. Brown, B. Rajashekar, and N. D. Pahnehkolaee. "Increasing Accuracy in Predicting Student Test Scores with Neural Networks using Domain Reduction Technique of Principal Component Analysis." *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2022.

[6] I. A. A. Amra, and A. Y. A. Maghari. "Students performance prediction using KNN and Naïve Bayesian." *2017 8th international conference on information technology (ICIT)*. IEEE, 2017.

[7] R. Sagardeep, and A.l Garg. "Predicting academic performance of student using classification techniques." *2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON)*. IEEE, 2017.

[8] P. Cortez, and A. M. G. Silva. "Using data mining to predict secondary school student performance.", 2008.

[9] J. Maurício, and G. Marques, "Predicting the Performance of Mathematics’ Students through Data Mining Techniques for Enhanced

Education Systems." *2021 1st Conference on Online Teaching for Mobile Education (OT4ME)*. IEEE, 2021.

[10] S. Kour, R. Kumar, and M. Guptam, "Analysis of student performance using Machine learning Algorithms." *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE, 2021.

[11] R. Ghorbani, and R, Ghousi, "Comparing different resampling methods in predicting students' performance using machine learning techniques." *IEEE Access* 8, 2020, pp. 67899-67911.

[12] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial." *Computer* 29.3, 1996, pp. 31-44.

[13] R. Gandhi, "Support vector machine—introduction to machine learning algorithms." *Towards Data Science* 7.06, 2018.

[14] A. Huda, et al., "Student performance prediction using support vector machine and k-nearest neighbor." *2017 IEEE 30th canadian conference on electrical and computer engineering (CCECE)*. IEEE, 2017.

[15] S. T. Ahmed, R. Al-Hamdani, and M. S. Croock, "Enhancement of student performance prediction using modified K-nearest neighbor." *TELKOMNIKA (Telecommunication Computing Electronics and Control)* 18.4, 2020, pp. 1777-1783.

[16] J. M. Brown, "Predicting math test scores using k-nearest neighbor." *2017 IEEE Integrated STEM Education Conference (ISEC)*. IEEE, 2017.

[17] B. N. Sari, "Prediksi Performa Akademik Siswa Pada Pelajaran Matematika Menggunakan Bayesian Networks dan Algoritma Klasifikasi Machine Learning.", 2017.

[18] S. K. Ghosh, and F. Janan. "Prediction of student's performance using random forest classifier." *Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Management, Singapore*. 2021.

[19] L. H. Alamri, et al. "Predicting student academic performance using support vector machine and random forest." *Proceedings of the 2020 3rd International Conference on Education Technology Management*. 2020.

[20] D. Engel, L. Hüttenberger, and B. Hamann, "A survey of dimension reduction methods for high-dimensional data analysis and visualization." *Visualization of Large and Unstructured Data Sets: Applications in Geospatial Planning, Modeling and Engineering-Proceedings of IRTG 1131 Workshop 2011*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2012.

[21] J. Mohamad-Saleh, and B. S. Hoyle, "Improved neural network performance using principal component analysis on Matlab." *International journal of the computer, the internet and Management* 16.2, 2008, pp. 1-8.

[22] I. T. Jolliffe, "Principal component analysis for special types of data", Springer New York, 2002.

[23] N. D. Pahnehkolaee, et al., "Increasing Accuracy in Predicting Student Test Scores with Regression Using Domain Reduction Technique of Principal Component Analysis." *2023 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2023.

[24] S. Karamizadeh, et al., "An overview of principal component analysis." *Journal of Signal and Information Processing* 4, 2020.

[25] S. Heo, and J. H. Lee, "Parallel neural networks for improved nonlinear principal component analysis." *Computers & Chemical Engineering* 127, 2019, pp. 1-10.

[26] L. R. Fabrigar, and D. T. Wegener, *Exploratory factor analysis*. Oxford University Press, 2011.

[27] B. Thompson, "Exploratory and confirmatory factor analysis: Understanding concepts and applications." *Washington, DC* 10694.000, 2004, pp. 3.

[28] T. A. Brown, "Confirmatory factor analysis for applied research." Guilford publications, 2015.

[29] J. B. Tenenbaum, Vin de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction." *science* 290.5500, 2000, pp. 2319-2323.

[30] L. Van der Maaten, and G. Hinton, "Visualizing data using t-SNE." *Journal of machine learning research* 9.11, 2008.