A University Admission Prediction System using Stacked Ensemble Learning

Sashank Sridhar
Department of Computer Science and Engineering
College of Engineering Guindy, Anna University
Chennai, India
sashank.ssridhar@gmail.com

Siddartha Mootha
Department of Computer Science and Engineering
College of Engineering Guindy, Anna University
Chennai, India
siddartha.mootha20@gmail.com

Santosh Kolagati
Department of Computer Science and Engineering
College of Engineering Guindy, Anna University
Chennai, India
ksantosh1399@gmail.com

***Abstract*— For an aspiring graduate student, shortlisting the universities to apply to is a difficult problem. Since an application is extremely dynamic, students often tend to wonder if their profile matches the requirement of a certain university. Moreover, the cost of applying to a university is extremely high making it critical that students shortlist universities based on their profile. A university admission prediction system is quite useful for students to determine their chances of acceptance to a specific university. The system could make use of data related to previous applicants to various universities and their admit or reject status. Earlier models of such prediction systems suffer from several drawbacks such as not considering important parameters like GRE (Graduate Record Exam) scores or research experience. Further, the accuracy reported by earlier models is also not sufficiently high. In this paper, a stacked ensemble model that predicts the chances of admit of a student to a particular university has been proposed. The proposed model takes into consideration various factors related to the student including their research experience, industry experience etc. Further, the system proposed has been evaluated against various other machine learning algorithms including other deep learning methods. It is observed that the proposed model easily outperforms all other models and provides a very high accuracy.**

Keywords*— *Admission, Deep Learning, Stacked Ensemble Model, Neural Network, Postgraduate studies, Prediction System, University, University Admission Prediction System

I. INTRODUCTION

Students aspiring to pursue higher studies are in a perennial state of confusion when it comes to deciding their university. Students have to consider numerous factors apart from the ranking of the university. One such factor is the acceptance rate of the university. Certain highly ranked universities accept only a very small percentage of students who apply to their programs. These are also universities that see a huge number of applicants. But the cost of the application for each university is very high eliminating students from applying to a large number of universities. Due to the high competition when it comes to applying for universities, and the cost associated with each application, students generally tend to apply to a handful of universities only. This is the most feasible option as well. Hence it would be extremely useful to students if there existed a prediction system that would rate their chances of getting into a university depending on their scores and academic background.

A university prediction system thus helps students narrow down their choices and ascertain their chances of getting into the university they desire. This also helps students make their own decisions instead of approaching counselors for getting university recommendations, or at least, helps them stay more informed of their chances.

University admission prediction can be achieved by considering the data related to a student and previous admits, then making a prediction based on this knowledge. A naïve approach to this is by training a model using basic machine learning algorithms. To achieve a higher degree of precision, deep learning methods can be adopted. In this paper, we propose a stacked ensemble classifier [10] that predicts an applicant's chances of getting an admit to a particular university.

Ensemble learning [11] is a type of machine learning method wherein multiple models are trained and combined to solve a specific problem. Traditionally, ensemble methods are used to combine "weak" models which prove to be strong when taken collectively. A stacked ensemble model, meanwhile, is an ensemble model that combines multiple machine learning models using a meta-machine learning model. The predictions from multiple sub-models are taken as the features of this meta-model which results in increased accuracy of the predictions. It was observed that the proposed method gave much better results in comparison with previous university admission prediction systems.

A dataset that provides valuable information of the applicants such as their undergraduate major, their TOEFL and GRE test scores, their publication count, their program of application, etc., has been considered. All these features combine to give an accurate prediction of the applicant's chances of being admitted to a specific university.

The rest of the paper is organized as follows: In Section II, we discuss the related works to university prediction systems and ensemble neural networks. Section III talks about ensemble neural networks in detail. Section IV describes the proposed method and provides a detailed explanation. In Section V, we compare the proposed method with other algorithms. Section VI and VII concludes the paper and discusses the future works.

II. RELATED WORKS

S.Fong et al. [1] proposed a hybrid model of Decision Tree Classifier and Neural Networks that predicts the chances of a student seeking admission in a specific university based on his academic merits and background. The proposed system was tested with the live data of 2400 Macau Secondary School students and considered thirteen attributes such as origin, major, rank in class, grades, etc. The results of the proposed

978-1-7281-6453-3/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on September 10,2022 at 09:58:25 UTC from IEEE Xplore. Restrictions apply.

method show that a hybrid classifier of decision trees and neural networks performs better than a single feed forward neural network or decision tree in forecasting whether a student is admitted or not in a particular university.

R. V. Mane et al. [2] designed a framework for predicting a student's admission to a particular college by using a hybrid combination of Association Rule Mining with Pattern Growth Approach. The attributes of the data source included student details such as name, gender, caste, address, 10th marks, 12th marks, score of Common Entrance Test, name of pre-college, name of admitted college, and branch. Once valid association rules are generated, the prediction is achieved by consequence constraint during the generation of association rules.

P.K. Binu et al. [9] proposed a framework for predicting a student's admission chances. The proposed framework has two modules i.e. A Hadoop MapReduce module to store the data and an Artificial Neural Network to predict the chances. The data collected had attributes such as state, rank, board, quota, etc. The system has not made use of academic credentials in predicting. The neural network has two input nodes, one hidden layer with two nodes, and an output layer with two nodes.

A.H.M. Ragab et al. [3] proposed a hybrid recommender system built on generic data mining approaches and knowledge discovery rules. The algorithm to predict uses historical data of colleges for students who were admitted/rejected. The system predicts the admission for a student based on the student's academic record and background information. The prototype was tested against live data provided by the On-Demand University Services, at King Abdulaziz University.

R. Swaminathan et al. [4] built a graduate recommender system for students aspiring to do their post-graduation in America. Data was scraped from Edulix for a set of 45 universities in the USA. On this scraped data, a feed-forward selection algorithm was run to select the ideal features to build the recommender system. The chosen features were the undergraduate university, GPA, GRE Score, and Research experience. The proposed method used the k-nearest neighbor, Support Vector Machine (SVM) and random forest algorithms to recommend universities. The SVM was the most efficient with 53.4% accuracy.

S. Girase et al. [5] built a hybrid recommendation engine using collaborative filtering, content-based filtering, and matrix factorization. A large dataset was created through a survey of over 100 colleges across India. The collected data was further split into two datasets, i.e. user interest dataset, and college rating dataset. Data from social media was scraped to build a user profile; this user profile was combined with the previously mentioned datasets and fed as input to the hybrid recommendation engine.

Hasan et al. [6] designed a recommender system that can help students choose the appropriate graduate school based on their current profile. A universal database is obtained and optimized, adding weights to every attribute such as GRE, TOEFL, IELTS, CGPA, etc. The proposed system using k-nearest neighbors to calculate top 'k' similar universities, based on this, the system recommends top 'N' universities to the user.

Dikhale et al. [7] proposed a framework that uses 160 entries of student data which include details such as gender, university, caste, merit number, and previous college. The authors make use of Naive Bayes and C4.5 algorithms to recommend universities. The disadvantage with the proposed system is that it doesn't take into consideration attributes such as GRE, GPA, IELTS, TOEFL, etc. which hold the most weightage for any application.

A. Baskota et al [8] proposed a hybrid recommender engine using the Multi-Class Support Vector Machine and KNN algorithm. The dataset was created by scraping Yocket and Edulix. Features such as GPA, GRE, TOEFL, IELTS, acceptance rate etc. were chosen using the feature selection algorithm. The authors recommended using C4.5, Naive Bayes, MLP and the hybrid SVM-KNN. The hybrid SVM-KNN showed the highest accuracy, close to 60%.

K. Zaamout et al. [11] proposed an ensemble based neural network framework. In this method, individual neural networks are trained and the output of each neural network is combined and used as the input for a larger neural network. By combining the forecast of every individual network, it has the potential to boost the classification of the entire system as a whole. The proposed system was tested against 4 readily available datasets and showed promising results.

III. Multi layer Perceptron and The Stacked Ensemble Model

A. The Multilayer Perceptron

The Perceptron is the most basic unit of a neural network modeled after a single neuron. A Multilayer Perceptron (MLP) is a collection of perceptrons (or neurons) connected to each other in layers [12]. The input data is fed to the input layer and the output is taken from the output layer. The layers between the input layer and the output layer are known as the hidden layers. The number of hidden layers and the number of neurons in each hidden layer can be varied according to convenience. Depending upon this, the complexity and accuracy of a model will change. A trial-and-error method is usually adopted to determine the ideal neural network architecture for a specific problem.

B. The Stacked Ensemble Model

Ensemble Learning is a type of machine learning method where multiple weak models are trained and their results are merged and used to solve a particular problem [14]. An MLP in itself isn't the most efficient neural network model for the university prediction system. There are several other deep learning methods that are more advanced. But when we stack multiple MLP models and create a stacked ensemble model, the results produced are exceptionally good. To achieve this, we initially train sub-models of the MLP and then train a meta-model using the predictions of the sub-models. The result is a superior deep learning model.

Ensemble learning is when multiple learning models, such as supervised algorithms capable of classification, are strategically fused to produce a superior model which is capable of improving the classification or prediction, or even reducing the chances of selecting a poor outcome.

There are two types of Ensemble Modeling:

1. Model Averaging Ensemble
2. Stacked Generalization

Model Averaging: The forecast of different models are combined to produce more accurate output. The prediction of

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on September 10,2022 at 09:58:25 UTC from IEEE Xplore. Restrictions apply.

the various models is merged. This could also be considered as the drawback of the model as there may be a scenario where one model performs better than the other yet there is equal weightage to both the models. To tackle this situation, each model is assigned weights. By assigning weights to every model, the model with better performance has more weightage than a weaker model. Finally, a weighted average is calculated and the final forecast is made.

Stacked Generalization: Stacked generalization, also known as stacking is an ensemble technique that combines the output of various classification algorithms to produce a more accurate forecast. The various models are combined using a meta-classifier.

The proposed method uses a stacked generalization.

Stacking has two phases: Level 0 and Level 1.

Level 0 corresponds to the individual sub-models. The training data is input to this level and the sub-models output the results. Level 1 is the combination model also called meta-learner. The meta-learner combines the outputs from the various sub-models in level 0 and produces another final output.

Figure 1 represents a general Stacking architecture.

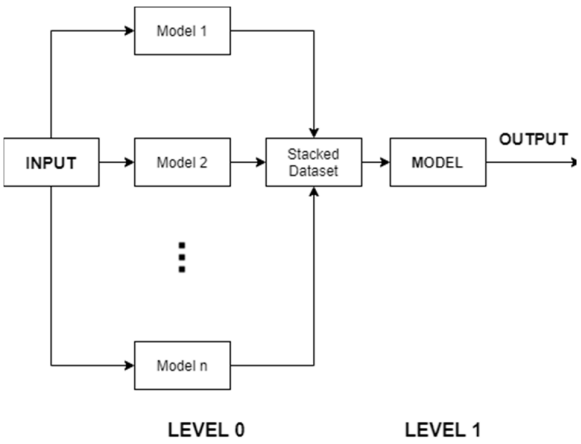

Fig. 1. General Stacking Architecture

IV. Implementation

A. Dataset

The dataset consists of the applicant scores such as TOEFL, IELTS, GRE, GMAT, CGPA, etc. R. Swaminathan et al. [4] built a web crawler to extract student details from Edulix for 45 universities shortlisted. The data scraping was achieved by using the python library 'BeautifulSoup'. The scraped data had a total of 22 features and 50, 000 samples. The dataset used is available on GitHub [15]. The attributes of the dataset post cleaning are listed in Table I.

B. Preprocessing

The dataset is preprocessed and cleaned before being input to the models. The first step is to normalize all the numerical values which include the scores and the CGPA data. Other attributes such as Major, University name, Department, Term, and Undergraduate university are also converted into a numerical format and normalized. This cleaning of the dataset is an essential step in the university prediction process. The cleaned dataset has a total of 35848 normalized samples

TABLE I. DATA ATTRIBUTES

Attribute Type	Attributes
Personal attribute	Name, userProfile
Program details	Program, Department, Term and year, University name
Scores	CGPA, Topper CGPA, TOEFL, GRE Q, GRE V, GRE A, IELTS
Undergraduate details	College name, Major, Specialization, CGPA Scale
Academic experience	Research experience, Journal publications, Conference publications
Industry experience	Industry experience, Intern experience

C. Sub-model training

This is the level 0 stage of the stacked ensemble model process. In this stage, the sub-models are trained. We consider five MLP models as our level 0 learners. Figure 2 demonstrates why five sub-models are ideal for Level 0.

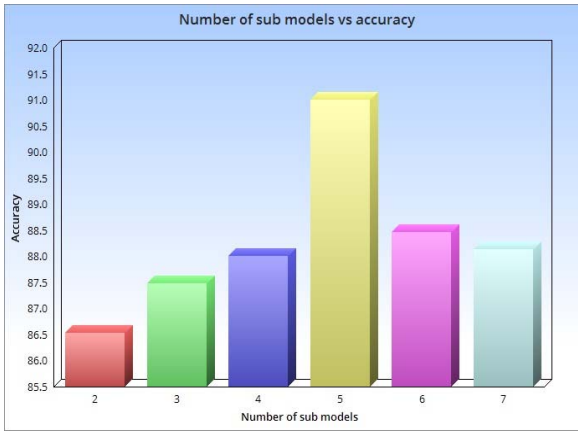

Fig. 2. Number of Sub-models vs accuracy

The input layer takes the training data attributes as input. The weights and biases of the sub-model are updated using the forward and backward propagation. Once this is done, the output layer will have an optimal activated value. The output layer gives the prediction. The output from this model is the chance of admit of the applicant to a specific university. Each sub-model is run for 500 epochs.

There are a total of 5 sub models, each having the same architecture. The input layer of the Multi-Layer Perceptron has 32 nodes, the input layer uses the 'ReLU' activation function. This is followed by the hidden layer, there are a total of 5 hidden layers each having 64, 128, 128, 64 and 32 nodes respectively. The 'ReLU' activation function is used in the hidden layer. The output layer forecasts a binary value i.e. 1 or 0. 1 if the student has been admitted to the specific college, 0 otherwise. The sigmoid activation function has been used.

All five of the sub-models are trained individually in the same manner. It is observed that the results from these predictions are "weak" or not sufficiently accurate. This is where the ensemble part comes in. In the next phase, a meta-learner utilizes the predictions from these five sub-models and

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on September 10,2022 at 09:58:25 UTC from IEEE Xplore. Restrictions apply.

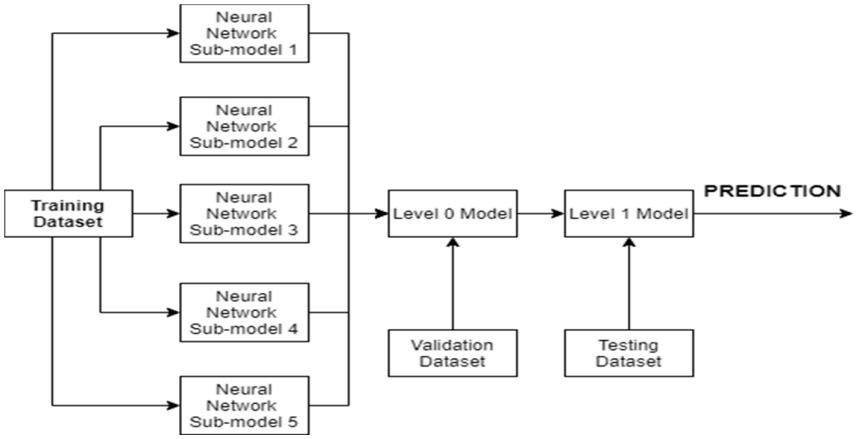

Fig. 3. Stacked Ensemble Model

undergoes training. The results of the meta-model will be a highly accurate prediction of the applicant's chances.

D. Stacked Ensemble Model

Figure 3 represents the architecture of the Stacked Ensemble Model. The level 1 model consists of 5 hidden layers. Figure 4 demonstrates why 5 hidden layers are optimal for the level 1 model.

If the dataset has X attributes which are fed to the network, and there are N sub models, the level 0 of the ensemble model will receive a total of X*N inputs. These outputs from the sub-models are concatenated to form the input vector for the level 1 model. Forward and backward propagation is done for the level 1 multilayer perceptron alone. The gradient descent is not propagated into the level 0 model since the level 0 and the level 1 networks act as two separate entities.

As a simple representation, let us take X to be the input of the level 0 model and Y to be the class of the input. W and b are the weights and biases of the model respectively. Each sub model in the level 0 will create a relation as follows.

$$Y' = W_x + b \tag{1}$$

The weight and bias are updated at every layer to achieve the least cost possible. Let the optimized output from the submodel be represented as Y′. The concatenated output from all the sub models can be represented as Y_x. This Y_x is given as input to the level 1 model.

$$Y_z = W_e Y_x + b \tag{2}$$

Here W_e and b represent the weights and the bias respectively.

The cost is optimized at every layer, and the least cost is selected. The training on the output probabilities of each sub model creates a two level optimized relationship between input X and output Y. This is the reason why ensemble models perform better than a simple neural network model.

V. Results And Discussions

Several metrics exist to evaluate machine learning models. For classification problems, a confusion matrix is generated. A confusion matrix is a 2×2 matrix with 4 possible values which correspond to the terms mentioned below.

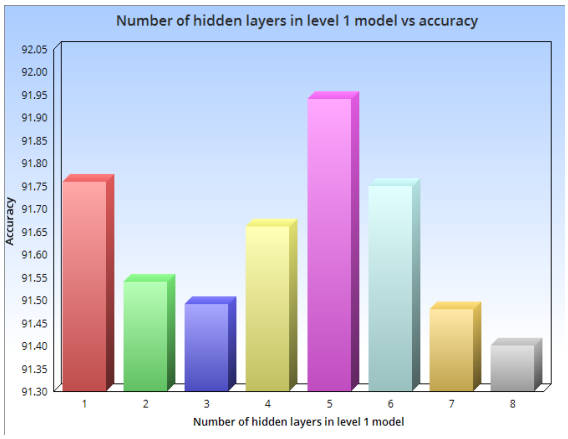

Fig. 4. Number of Hidden Layers in Level 1 Model vs Accuracy

i. True Positive (TP) – A sample the model correctly predicts to be the positive class.

ii. True Negative (TN) – A sample the model correctly predicts to be the negative class.

iii. False Positive (FP) – A sample the model predicts as positive when it is negative.

iv. False Negative (FN) – A sample the model predicts as negative when it is positive.

With the help of a confusion matrix, it is possible to define other evaluation metrics such as Precision, Recall, F1-Score, and Accuracy. These metrics can be calculated as mentioned below.

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

$$F1\ Score = 2\frac{Precision \times Recall}{Precision+Recall} \tag{5}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{6}$$

The proposed ensemble neural network is evaluated by comparing it to other supervised algorithms such as Decision Trees, Random Forest, K-Nearest Neighbor, Naive Bayes Classifier, Logistic Regression, Support Vector Machine

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on September 10,2022 at 09:58:25 UTC from IEEE Xplore. Restrictions apply.

(SVM), Linear Discriminant Analysis and Quadratic Discriminant Analysis.

A. Decision Tree

A decision tree is a supervised learning algorithm which is modelled around iterative binary recursive partitioning [13]. A decision tree has a root node, edges and leaf nodes. A leaf node represents a prediction/output. The data is recursively split into partitions for various scenarios. A decision tree is an ensemble technique, which acts as the building block for Random Forest classification. There are various Decision Tree algorithms such as ID3, C4.5, CART etc. This paper makes use of the CART algorithm available in the sci-kit learn library in python to predict whether a student will get accepted or rejected by a particular university. The algorithm reported an accuracy of 65.5%.

B. Random Forest

Random forest is an ensemble learning method which is predominantly used for classification purposes [17]. A random forest consists of a large number of decision trees; each tree predicts a particular class, the class with the majority votes is chosen as the output of the ensemble classifier. A random forest has been created using 2, 4, 8, 16, 32, 64, 100 and 200 decision trees. As can be seen from Figure 5, the model was found to have the highest accuracy at 30 trees. The Random forest model reports an accuracy of 62.5% which can be further improved.

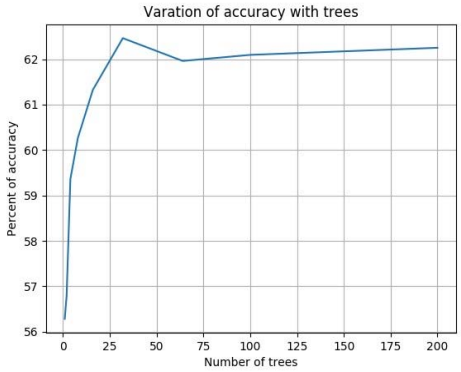

Fig. 5. Random forest variation of accuracy with trees

C. KNN (K-Nearest Neighbours)

The K-Nearest Neighbors is a versatile algorithm that can be used for classification purposes [19]. All the possible class attributes are mapped on an N-dimensional feature space. K is the number of nearest neighbors chosen for every scenario. To classify a new sample, the algorithm makes use of the K value; it finds the K-nearest neighbor, each neighbor pertaining to a particular class i.e. accept or reject, and the class with the majority votes is classified as the output. The nearest neighbors can be found using various distance measures such as Euclidean, Manhattan etc. For our university prediction system, the model was run for various values of K, at 15 neighbors and the system gave an accuracy of 57%, as represented by Figure 6.

D. Naive Bayes Classifier

The Naive Bayes Classifier is a probabilistic machine learning algorithm based on the popular Bayes Theorem. Naive Bayes Classifier [16] is a supervised learning algorithm that uses conditional probability to predict the class of a particular sample. This paper makes use of the 'GaussianNB' classifier available in the sci-kit learn library in python. The Naive Bayes Classifier provides an accuracy of 53%.

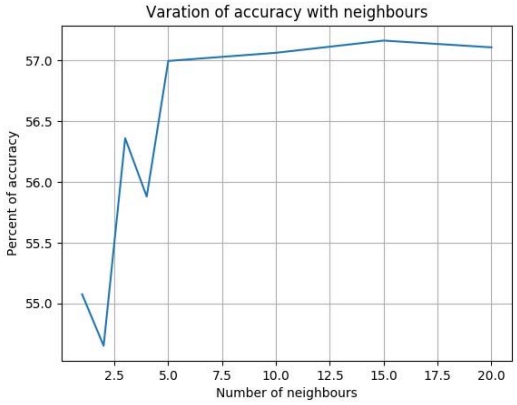

Fig. 6. KNN variation of accuracy with neighbors

E. Logistic Regression

Logistic Regression is a statistical model that uses the given set of attributes along with the class label to classify the sample [21]. The algorithm computes a linear combination of the given input and feeds it to a logistic function. The logistic function is able to classify the sample based on the linear computation that is computed. The algorithm has an accuracy of 56% for the university prediction system.

F. SVM (Support Vector Machine)

A Support Vector Machine is a classification algorithm that makes use of a hyperplane to classify data points [20]. The algorithm constructs a hyperplane in a multidimensional space that distinguishes the two class labels: accept and reject. Via the hyperplane, the margin between the two class labels is maximized. The linear SVM kernel is used in this paper. The SVM gives an accuracy of 55% in predicting admits.

G. Linear and Quadratic Discriminant Analysis

LDA (Linear Discriminant Analysis) and QDA (Quadratic Discriminant Analysis) are statistical supervised learning algorithms [18]. Both methods make use of a covariance matrix which provides information about the scatter within a particular dataset. Scatter is the amount of spread that is there in a specific dataset. LDA and QDA are also used for dimensionality reduction. With the help of the covariance matrix and information about the scatter, the predictions are made. The primary difference between both the algorithms is that in QDA, there is no guarantee that the covariance matrix is identical for every class label. LDA and QDA report an accuracy of 56% and 53% respectively.

Table II. compares the precision, recall, F1-score, and accuracy for the models. Figure 7 is a visual representation of the accuracy of the various models. It can be observed that the proposed ensemble neural network provides a much higher accuracy than all the other supervised learning algorithms. Since the level 1 ensemble model is trained using the output probabilities of each sub-model, it creates a two level optimized relationship between the output of the level 0 ensemble model and the input of the level 1 ensemble model. This is the primary reason why the ensemble model performs better than a traditional neural network.

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on September 10,2022 at 09:58:25 UTC from IEEE Xplore. Restrictions apply.

TABLE II. COMPARISON OF ALGORITHMS

S No	Model	Precision	Recall	F1-Score	Accuracy
1	Decision Tree	0.65	0.65	0.65	65.5%
2	Random Forest	0.63	0.63	0.63	62.5%
3	KNN	0.57	0.57	0.57	57%
4	Naive Bayes	0.54	0.53	0.47	53.3%
5	Logistic Regression	0.54	0.54	0.54	55.88%
6	SVM	0.55	0.55	0.55	55%
7	LDA	0.56	0.56	0.56	56.1%
8	QDA	0.56	0.55	0.52	53.6%
9	Basic NN	0.56	0.56	0.56	56%
10	Ensemble NN	0.91	0.91	0.91	91%

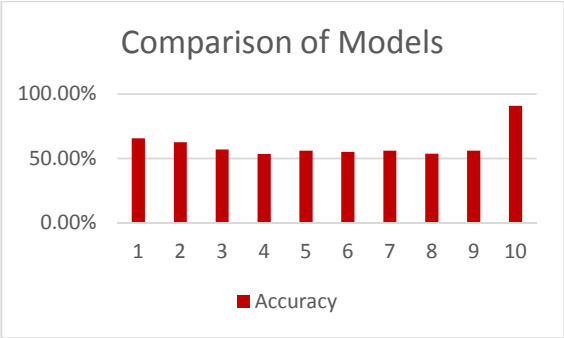

Fig. 7. Comparison of Models

VI. CONCLUSION

With an increasing number of students looking to pursue a degree, the domain of university admission prediction gains more relevance. In this paper, an effective method has been proposed to predict the chances of a student being admitted to a specific university. In addition, we have compared the performance of various machine learning algorithms to the proposed method in predicting admits. It is observed that the proposed method provides the best performance with an accuracy of 91%.

VII. FUTURE WORK

In the future, the model can be improved as we gain more data about students. Details regarding the applicant's Statement of Purpose essay and Letters of Recommendation (a score can be assigned to rate these) can be used to improve the prediction accuracy. An alternative is to use Natural Language Processing methods to evaluate the essays and letters. In addition to this, the weightage given to the features can be varied according to past trends.

ACKNOWLEDGMENT

We are thankful to Dr. V. Vetriselvi, Associate Professor, Department of Computer Science and Engineering, College of Engineering Guindy for her valuable contributions to this research.

REFERENCES

[1] S. Fong, Y. Si and R. P. Biuk-Aghai, "Applying a hybrid model of neural network and decision tree classifier for predicting university admission", 2009 7th International Conference on Information, Communications and Signal Processing (ICICS), Macau, pp. 1-5,2009

[2] R. V. Mane and V. R. Ghorpade, "Predicting student admission decisions by association rule mining with pattern growth approach", 2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT), Mysuru, pp. 202-207, 2016

[3] A. H. M. Ragab, A. F. S. Mashat and A. M. Khedra, "HRSPCA: Hybrid recommender system for predicting college admission", 2012 12th International Conference on Intelligent Systems Design and Applications (ISDA), Kochi, pp. 107-113, 2012

[4] R. Swaminathan, J. M. Gnanasekaran, S. Krishnakumar, A. Suresh kumar, "University Recommender System for Graduate Studies in USA", unpublished

[5] S. Girase, V. Powar and D. Mukhopadhyay, "A user-friendly college recommending system using user-profiling and matrix factorization technique", 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, pp. 1-5, 2017

[6] M. Hasan, S. Ahmed, D. M. Abdullah and M. S. Rahman, "Graduate school recommender system: Assisting admission seekers to apply for graduate studies in appropriate graduate schools", 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV), Dhaka, pp. 502-507, 2016

[7] N. Dikhale, H. Shrivastava, A. Dudhane, and U. Gholap, "College Recommendation System", In: NCPCI, pages 112– 114, 2016.

[8] A. Baskota and Y. Ng, "A Graduate School Recommendation System Using the Multi-Class Support Vector Machine and KNN Approaches", 2018 IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, pp. 277-284, 2018

[9] P. K. Binu, A. Chandran and M. Rahul, "A Cloud-Based Data Analysis and Prediction System for University Admission", 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur,Kerala, India, pp. 1327-1332, 2019

[10] K. Zaamout and J. Z. Zhang, "Improving classification through ensemble neural networks", 2012 8th International Conference on Natural Computation, Chongqing, pp. 256-260, 2012

[11] F. Huang, G. Xie and R. Xiao, "Research on Ensemble Learning", 2009 International Conference on Artificial Intelligence and Computational Intelligence, Shanghai, pp. 249-252, 2009

[12] P. Marius, V. Balas, L. Perescu-Popescu and N. Mastorakis, "Multilayer perceptron and neural networks", In: WSEAS Transactions on Circuits and Systems. 8, 2009

[13] B. N. Patel, S. G. Prajapati, and K. I. Lakhtaria, "Efficient Classification of Data Using Decision Tree", In: Bonfring International Journal of Data Mining, 2012

[14] D. Opitz, and R. Maclin. "Popular Ensemble Methods: An Empirical Study", In: Journal of Artificial Intelligence Research 11, 169–198. Crossref. Web, 1999

[15] J. M. Ganesan, "Recommendation System for Graduate Level Universities in the United States", https://github.com/joemanley201/universityRecommendationSystem/blob/master/scraper/univJSON/allUsersFinal.csv

[16] I. Rish, "An Empirical Study of the Naïve Bayes Classifier", In: IJCAI 2001 Work Empir Methods Artif Intell. 3, 2001

[17] L. Breiman, "Random Forests", Machine Learning 45, 5–32, 2001

[18] B. Ghojogh, and M. Crowley, "Linear and Quadratic Discriminant Analysis: Tutorial", In: Tutorial. arXiv preprint arXiv:1906.02590, 2019

[19] P. Cunningham and S. Delany, "k-Nearest neighbour classifiers", In: Multiple Classifier Systems, Prague, Czech Republic, pp. 1-17, 2007

[20] N. Cristianini and J. Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods", U.K., Cambridge:Cambridge Univ. Press, 2000.

[21] C. J. Peng, K. L. Lee and G. M. Ingersoll, "An Introduction to Logistic Regression Analysis and Reporting" In: The Journal of Educational Research, 96:1, 3-14, 2002

Authorized licensed use limited to: ANNA UNIVERSITY. Downloaded on September 10,2022 at 09:58:25 UTC from IEEE Xplore. Restrictions apply.