Study of Machine Learning Models for IoT based Efficient Classroom Usage

Yugay, O., Yerashenia, N. and Budimir, D.

https://doi.org/10.1007/978-3-031-60994-7_21

Study of Machine Learning Models for IoT based Efficient Classroom Usage

Olga Yugay[1,3], Natalia Yerashenia[2] and Djuradj Budimir[3]
[1]Business Information Systems, SOLTE, Westminster University in Tashkent,
Tashkent, Uzbekistan
[2]Software Systems Engineering Research Group
[3]Wireless Communication Research Group
School of Computer Science and Engineering,
University of Westminster,115 New Cavendish Street, London, W1W 6UW, UK
oyugay@wiut.uz, N.Yerashenia3@westminster.ac.uk, d.budimir@westminster.ac.uk

***Abstract* — This paper presents performance analysis and comparison of machine learning algorithms for future use in a smart campus framework. The following error rates, such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Square Error (MSE) and R squared error are considered for models such as Random Forest (RF), Multiple Linear Regression (MLR), Decision Tree Regression (DTR), Support Vector Regression (SVR), Polynomial Regression (PR), Generic Predictive Computation Model (GPCM). The investigation how to reduce the processing time for the algorithms is presented. The following error rates such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Square Error (MSE) are considered for Random Forest, Multiple Linear Regression, Decision Tree Regression, Support Vector Regression, Polynomial Regression models and Machine Learning tools taken from Use Cases of Generic Predictive Computation Model (GPCM) are partially applied. Testing with our arbitrary data will be conducted. A lower error rate for selected algorithms with reduced number of parameters (5 parameters) as opposed to 11 parameters is achieved.**

***Keywords* —** AI, ML, Classroom usage, Prediction, IoT sensors, Smart campus.

1. INTRODUCTION

In modern educational institutions the attendance may vary or even go down due to factors like time-of-day [1, 2, 3], availability of online content [1, 2], lecturer engagement [1, 2, 3], attendance policies and monitoring [3]. As a result, there is a need for *optimising* the use of higher education resources for their *users*. The use of Internet of Things (IoT) tools and Machine Learning (ML) algorithms can contribute to a more efficient timetable by predicting the attendance and adjusting the schedule. There are a number of suitable algorithms/models to predict attendance that can be updated. This paper suggests prioritising and reviewing the number of parameters used in selected ML algorithms/models and Generic Predictive Computation Model [4], which results in lower error rates and increases their efficiency.

The definition of timetabling and review of its types is well categorised by C.B. Mallari et. al. [5] as (a) *school timetabling* (Ahmed et al., [6]; Beligiannis et al., [7]; Birbas et al., [8] in [5]), (b) *course timetabling* (Yasari et al., [9]; Rezaeipanah et al., [10]; Algethami & Laesanklang, [11] in [5]) and (c) *examination/coursework timetabling* (Al-Yakoob et al., [12]; Burke & Bykov, [13]; Leite et al., [14]; Abou Kasm et al., [15] in [5]).

Overall, there are various methods that can contribute to improving the speed and efficiency of timetabling algorithms. There is a group of research [1, 6-12] work from the perspective of reviewing and improving the algorithms and generating the timetable. Another group of them identified that possibility in a more efficient and optimal allocation of classrooms by attempting to focus on predicting the potential classroom occupancy based on machine learning (ML) algorithms [1, 2, 8]. The latter is a more efficient approach in terms of managing the resources since the research shows that attendance keeps falling due to diverse demands of student time, growing student employment, and easy access to online content. [2, 3] As a result, there is a growing university pressure to optimise the use of its resources, in particular the classrooms, and the associated operating costs. With a carefully adjusted timetable that can be predicted based on IoT sensor collected attendance data, the university may achieve needed savings in operating costs. Before modelling the predictive framework and testing actual data, it was decided to experiment

with the arbitrary data on a selected number of ML algorithms/models and choose the most efficient algorithms/models. The choice of models/algorithms includes Decision Tree Regression, Multiple Linear Regression, Polynomial Regression, Random Forest Regression, Support Vector Regression, partially used Machine Learning tools taken from use cases of Generic Predictive Computation Model (GPCM) [4]. The details of findings are described in further paragraphs.

RELATED WORKS

Due to its multidisciplinary approach, the Internet of Things (IoT) has revolutionised traditional educational paradigms, enabling efficient and productive educational applications and services [1] [16]. Over the past two decades, IoT networks and sensor networks have been successfully applied in various educational applications, such as using Artificial Intelligence (AI) to optimise classroom usage and predict room occupancy using Wi-Fi Soft Sensors [6] and [17]. IoT sensors are utilised to measure real-time class attendance, allowing to collect necessary data. AI algorithms are then employed to predict attendance based on the collected data and allocate rooms optimally for courses [8]. This exemplifies how a smart campus can effectively optimise its resources. As a solution, IoT and AI applied in data analytics can be used for resolving these problems. Due to its multidisciplinary approach, the Internet of Things (IoT) has been innovative in revolutionising many aspects of traditional educational paradigms so that educational applications and services can be obtained with high efficiency and productivity. In the last two decades, IoT networks and sensor networks have been applied for various education applications, such as Artificial Intelligence (AI) for optimising classroom usage or predicting room occupancy using Wi-Fi Soft Sensors [1, 6, 7, 8].

In particular, there are multiple studies on how IoT can offer benefits via location-based user applications and monitor the use of space [18, 19] in [20]. According to Valks, B., et. al., [20] most types of IoT applications tend to prefer a level of granularity that is at the room level or higher. The exceptions, however, are found on user flows at floor and building levels. The objective of this paper is to explore AI solutions for future modelling an IoT-based predictive smart campus framework, and focus on Machine Learning (ML) algorithms to contribute to a more efficient timetable by predicting the attendance. Specifically, the paper examines and compares the performance of various regression algorithms, such as Decision Tree Regression, Multiple Linear Regression, Polynomial Regression, Random Forest Regression, Support Vector Regression [2, 18], and Generic Predictive Computation Model (GPC) [4]. The paper also seeks analysis of the implications of the proposed method on the dataset use case [2]. By comparing the available algorithms and models, the study aims to identify the most effective approach for predicting attendance with minimal error rates.

2. PROBLEM SETUP

RESEARCH METHODOLOGY

In the first step of our research methodology, which follows the Knowledge Discovery Database (KDD) Process described by Ahmad Sabri et al, where in the first step we *prepare the data*. We split the data 30% test size and 70% train size. The next step is *feature engineering*, which is focused on normalised attendance and numeric matrix. In our case, ***normalised attendance*** is the ratio of maximum classroom occupancy to enrolment count. More detailed explanation on it in the next section.

Score based categorical feature engineering was used for MLPClassifier. The next step was the *feature selection*. The original study dataset contains 18 features. Less features were considered for the given experiment. The key moment about this is to **prioritise and review the number of parameters** used in the corresponding models. Finally, for modelling and evaluation the following models were considered: Decision Tree Regression, Multiple Linear Regression, Polynomial Regression, Random Forest Regression, Support Vector Regression, partially used Machine Learning tools taken from use cases of Generic Predictive Computation Model (GPCM) [1]. In the case of GPCM, the neural networks MLPClassifier was applied.

3. DATA PREPARATION AND PRE-PROCESSING

For the initial algorithm test the arbitrary data from the use case of the partially based on dataset [2] was used, see Figure 1. It followed the process of discovering useful knowledge from a collection of data - Knowledge Discovery in Databases (Ahmad Sabri et al, 2019). [12]

The data has the following arbitrary parameters. For more information, enter the following link: https://github.com/olga-yu/ML_models_for_efficient_classroom. To prepare data for the analysis data transformation was performed such as selection and pre-processing. Original data sample is presented in Figure 1. After data transformation steps, the data was transformed to look like in the sample presented in Figure 2. Additional attributes were added after the pre-processing, this include **normalised attendance1** and **normalised attendance2**. Normalised attendance1 output was calculated based on the *number of students attended* and equal to 0 to the *students enrolled* and equal to 1. It calculated the ratio of maximum classroom occupancy to enrolment count. Another pre-processing step that has been completed is categorisation of the users into 3 categories. Normalised attendance2 output is created based on conditions to meet this value. Such a problem with more than two classes is often called a multi-class classification problem.

Normalized_attendance and normalized_attendance2 are calculated according to the following formulas:

normalised_attendance = attendance/enrolment
normalised_attendance2=categorized numbered/normalised_attendance

The normalised attendance2 is sorted according to the following criteria: if the normalised attendance2 value is less than 0.3 then output is 0, if the normalised attendance2 value is less or equal to 0.6 then output would be 1, finally for normalised attendance2 greater than 0.6 then it is 2. In addition, the attendance column has no missing values, no duplicates.

	year	semeste	week	date	day	time_of_c	start_time	end_time	room_nan	class_type	faculty	school	joint	status	degree	enrollmer	class_dura	attendan
1	2017	T2	9	9/21/2017	thu	morning	9:00:00	10:00:00	Mathews	Lecture	Faculty of	School of	FALSE	Open	Undergrad	452	1:00:00	95
2	2017	T2	3	8/9/2017	wed	morning	9:00:00	10:00:00	Mathews	Lecture	Faculty of	Sch Mathe	FALSE	Open	Undergrad	447	1:00:00	136
3	2017	T2	9	9/21/2017	thu	afternoon	13:00:00	14:00:00	Mathews	Lecture	Faculty of	School of	FALSE	Open	Undergrad	419	1:00:00	93
4	2017	T2	3	8/8/2017	tue	morning	11:00:00	13:00:00	Mathews	Lecture	Faculty of	Sch Mathe	FALSE	Open	Undergrad	381	2:00:00	181
5	2017	T2	3	8/8/2017	tue	evening	16:00:00	18:00:00	Mathews	Lecture	Faculty of	School of	FALSE	Open	Undergrad	459	2:00:00	182
6	2017	T2	10	10/5/2017	thu	afternoon	13:00:00	14:00:00	Mathews	Lecture	Faculty of	School of	FALSE	Open	Undergrad	419	1:00:00	87
7	2017	T2	10	10/6/2017	fri	morning	9:00:00	10:00:00	Mathews	Lecture	Faculty of	School of	FALSE	Open	Undergrad	419	1:00:00	57
8	2017	T2	9	9/19/2017	tue	evening	16:00:00	18:00:00	Mathews	Lecture	Faculty of	School of	FALSE	Open	Undergrad	459	2:00:00	103

Figure 1. The sample from original dataset [5]

	year	week	date	day	time_of_c	room_nan	class_type	faculty	school	joint	status	degree	enrollmer	class_dura	attendanc	class_type	attendanc	date-year	date-mon	date-day	normalize	normalized_attendance2
0	2017	8	9/21/2017	2	2	2	0	2	16	0	4	2	452	1	95	3	1	2017	9	21	0.210177	0
1	2017	2	8/9/2017	4	2	2	0	5	8	0	4	2	447	1	136	4	2	2017	8	9	0.304251	1
2	2017	8	9/21/2017	2	0	2	0	5	15	0	4	2	419	1	93	3	1	2017	9	21	0.221957	0
3	2017	2	8/8/2017	3	2	2	0	5	8	0	4	2	381	2	181	4	2	2017	8	8	0.475066	1
4	2017	2	8/8/2017	3	1	2	0	2	16	0	4	2	459	2	182	4	2	2017	8	8	0.396514	1
5	2017	9	10/5/2017	2	0	2	0	5	15	0	4	2	419	1	87	3	1	2017	10	5	0.207637	0

Figure 2. The sample from processed dataset

It is common for classification models to predict a continuous value as the probability of a given example belonging to each output class. The probabilities can be interpreted as the likelihood or confidence of a given example belonging to each class. A predicted probability can be converted into a class value by selecting the class label that has the highest probability.

- For example, a specific email of text may be assigned the probabilities of 0.1 as being “spam” and 0.9 as being “not spam”. We can convert these probabilities to a class label by selecting the “not spam” label as it has the highest predicted likelihood.

4. RESULTS

The testing and prediction were conducted on arbitrary data. As a result of the given research, we managed to work on the tool for important feature selection that enables us **to prioritise and review the number of parameters** used in the algorithms/models listed below, resulting in lower error rates thus being more efficient. Below the results are discussed by referring to the corresponding figures and tables. For example, feature importance can be visually observed on the graphic plots, and it can be clearly seen that in SVR, the first 4 have clear advantages over others: class_type, class_duration, degree, and joint (See Figure 3).

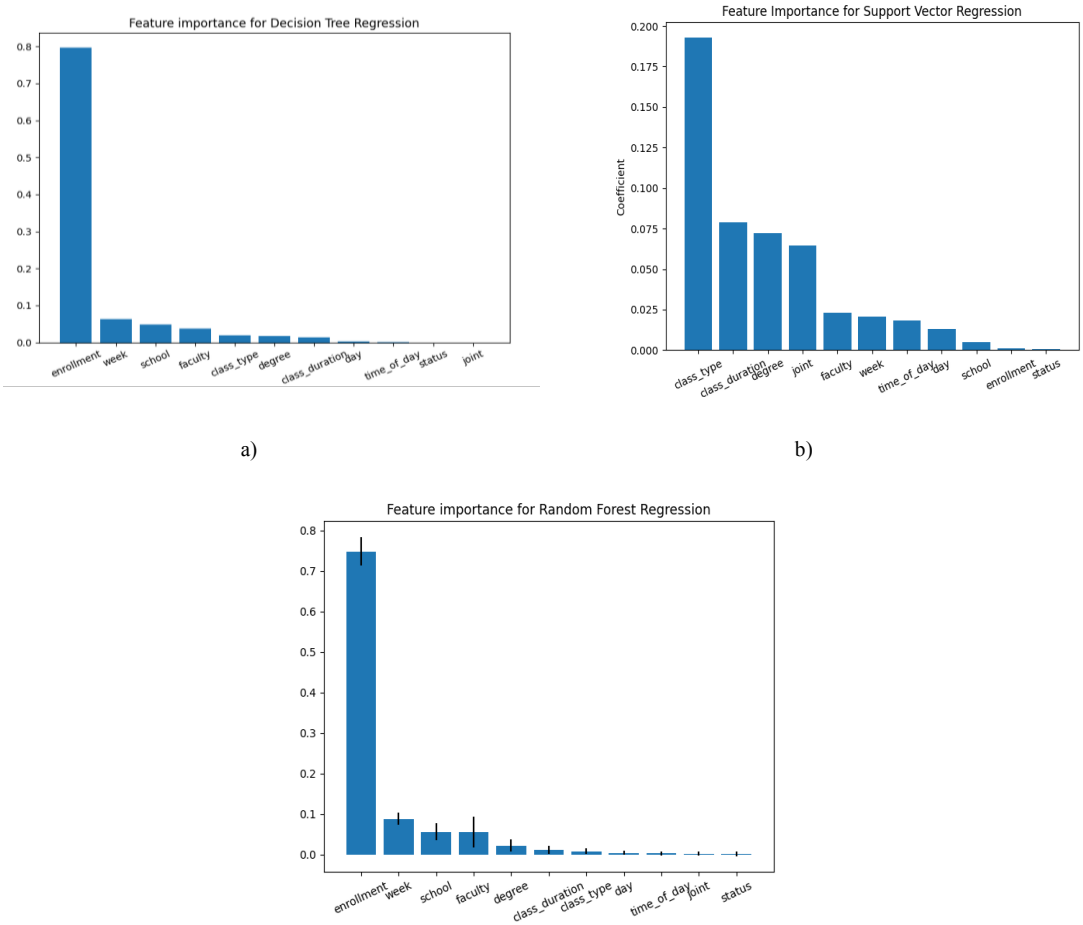

Figure 3. Feature importance compared in various algorithms: a) Decision tree, b) Support Vector Regression, and c) Random Forest Regression

The Table 2a shows values highlighted in *italic* with the following errors such as RMSE, MAE, and MSE, where the standard random forest (SRF) regression algorithm outperforms the standard decision tree (SDT) algorithm in an experiment with 11 parameters. The 11 parameters present 'week', 'day', 'time_slot', 'class_type', 'faculty', 'school', 'joint', 'status', 'degree', 'enrollment', and 'class_duration'. The data and parameters are based on our arbitrary data and applied random forest algorithm with help of Sklearn using Python programming language and achieved results presented in the Table 2a. The main goal of research is to reduce the possible error rate by reviewing the 11 parameters used in Table 2a, such as shown in Table 2b has values highlighted in bold where RMSE, MAE, R square, MSE values were achieved from experimenting with **5 most important parameters** in corresponding algorithms.

These values outperform the standard 11 parameters implemented in Table 2a. The original 11 parameters include: **'week', 'day', 'timeslot', 'class_type', 'faculty', 'school', 'joint', 'status', 'degree', 'enrolment', 'class_duration'.** The 5 parameters to focus on include: **'week', 'school', 'enrolment','day', 'faculty**'.

Table 2. Experiment with 11 parameters and 5 most important parameters

(a) Experiment with 11 parameters: 'week', 'day', 'time_of_day', 'class_type', 'faculty', 'school', 'joint', 'status', 'degree', 'enrolment', and 'class_duration' as listed in key paper					**(b) Experiment with *5 most important parameters*: For example: 'week', 'school', 'enrolment','day', 'faculty' The parameters may vary from model to model**				
	2017 test set (testing)					**2017 test set (testing)**			
	RMSE	**MAE**	**R squared**	**MSE**		**RMSE**	**MAE**	**R squared**	**MSE**
Multiple Linear Regression	0.16	0.12	65.25	0.02	Multiple Linear Regression	0.18	0.15	**56.11**	0.03
Random Forest Regression	***0.14***	***0.11***	76.73	***0.02***	Random Forest Regression	**0.13**	**0.10**	81.55	**0.01**
Decision Tree Regression	0.15	0.12	72.81	0.02	Decision Tree Regression	0.15	**0.11**	**71.86**	0.02
Support Vector Regression	0.17	0.13	64.9	0.03	Support Vector Regression	0.22	0.17	**38.29**	0.04
Polynomial Regression	0.15	0.11	69.49	0.02	Polynomial Regression	0.16	0.13	**64.55**	0.02
GPCM	0.57	0.30	-	0.33	GPCM	0.60	0.32	-	0.36

5. CONCLUSION

In summary, the machine learning algorithms were analysed and compared in their performance for future use in smart campus framework in this paper. The following error rates, such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Square Error (MSE) have shown similar results but mostly the R squared error has shown better results with reduced number of features for models like Random Forest, Multiple Linear Regression, Decision Tree Regression, Support Vector Regression, Polynomial Regression, Generic Predictive Computation Model (GPCM). Reducing the number of features can reduce the processing time. The following error rates such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Square Error (MSE) have been considered for Random Forest, Multiple Linear Regression, Decision Tree Regression, Support Vector Regression, Polynomial Regression models) and partially used Machine Learning tools taken from Use Cases of Generic Predictive Computation Model (GPCM). The validation of investigations based on testing with our arbitrary data showed that the algorithms achieved a lower error rate, when a smaller number of parameters were used.

ACKNOWLEDGEMENT

This work was supported by a WIUT-UOW Research collaboration funded project under grant RCF2022-003.

REFERENCES

[1] Mohottige I.P, Moors, T. (2018). Estimating Room Occupancy in a Smart Campus using WiFi Soft Sensors. Local Computer Networks. doi: https://doi.org/10.1109/lcn.2018.8638098

[2] Thanchanok Sutjarittham, "Data-Driven Monitoring and Optimization of Classroom Usage in Smart Campus" 2018.

[3] Elisabeth Moores, Gurkiran K. Birdi & Helen E. Higson (2019) Determinants of university students' attendance, Educational Research, 61:4, 371-387, DOI: 10.1080/00131881.2019.1660587

[4] Yerashenia, N., Chan You Fee, D. and Bolotov, A. (2022). Developing a Generic Predictive Computational Model using Semantic data Pre-Processing with Machine Learning Techniques and its application for Stock Market Prediction Purposes. 24th IEEE International Conference on Business

Informatics (IEEE CBI 2022). Amsterdam 15 - 17 Jun 2022 IEEE . https://doi.org/10.1109/cbi54897.2022.00013
[5] Mallari, C.B., San Juan, J.L. and Li, R., (2023). The university coursework timetabling problem: An optimization approach to synchronizing course calendars. Computers & Industrial Engineering, 184, p.109561.
[6] Ahmed, L.N., Özcan, E., Kheiri, A. (2015). Solving high school timetabling problems worldwide using selection hyper-heuristics. Expert Systems with Applications, 42(13), 5463-5471.
[7] Beligiannis, G.N., Moschopoulos, C.N., Kaperonis, G.P., Likothanassis, S.D. (2008). Applying evolutionary computation to the school timetabling problem: The Greek case.
[8] Birbas, T., Daskalaki, S., Housos, E. (2009). School timetabling for quality student and teacher schedules. Journal of Scheduling, 12(2), 177-197. doi: 10.1007/s10951-008-0088-2.
[9] Yasari, Peyman, et al. "A Two-Stage Stochastic Programming Approach for a Multi-Objective Course Timetabling Problem with Courses Cancelation Risk." Computers & Industrial Engineering, vol. 130, Apr. 2019, pp. 650–660, https://doi.org/10.1016/j.cie.2019.02.050.
[10] Rezaeipanah, Amin, et al. "A Hybrid Algorithm for the University Course Timetabling Problem Using the Improved Parallel Genetic Algorithm and Local Search." Applied Intelligence, 19 Aug. 2020, https://doi.org/10.1007/s10489-020-01833-x. Accessed 20 Aug. 2020
[11] Algethami, H., and W. Laesanklang. "A Mathematical Model for Course Timetabling Problem with Faculty-Course Assignment Constraints." IEEE Access, vol. 9, 2021, pp. 111666–111682, https://doi.org/10.1109/access.2021.3103495.
[12] Al-Yakoob, Salem M., et al. "A Mixed-Integer Mathematical Modeling Approach to Exam Timetabling." Computational Management Science, vol. 7, no. 1, 1 Dec. 2007, pp. 19–46, https://doi.org/10.1007/s10287-007-0066-8.
[13] Burke, E. K., Elliman, D. G., Ford, P. H., & Weare, R. F. (1998). The university coursework timetabling problem: An optimization approach to synchronizing course calendars. The Journal of the Operational Research Society, 49(7), 724-738.
[14] Leite, Nuno, et al. "A Fast Simulated Annealing Algorithm for the Examination Timetabling Problem." Expert Systems with Applications, vol. 122, May 2019, pp. 137–151, https://doi.org/10.1016/j.eswa.2018.12.048. Accessed 10 Nov. 2019.
[15] Abou Kasm, Omar, et al. "Exam Timetabling with Allowable Conflicts within a Time Window." Computers & Industrial Engineering, vol. 127, Jan. 2019, pp. 263–273, https://doi.org/10.1016/j.cie.2018.11.037. Accessed 18 May 2021.
[16] Kostuch, P. (2005) "The university course timetabling problem with a three-phase approach," Practice and Theory of Automated Timetabling V, pp. 109–125. Available at: https://doi.org/10.1007/11593577_7
[17] Larabi-Marie-Sainte, S., Jan, R., Al-Matouq, A. and Alabduhadi, S., 2021. The impact of timetable on student's absences and performance. Plos one, 16(6), p.e0253256.
[18] Sutjarittham, T., Gharakheili, H.H., Kanhere, S.S. and Sivaraman, V. (2019). Experiences with IoT and AI in a Smart Campus for Optimizing Classroom Usage. IEEE Internet of Things Journal, pp.1–1. doi: https://doi.org/10.1109/jiot.2019.2902410
[19] Sutjarittham, T., Gharakheili, H. H., Kanhere, S. S., & Sivaraman, V. (2018). Realizing a smart university campus: Vision, architecture, and implementation. *2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. https://doi.org/10.1109/ants.2018.8710084
[20] Valks, B., Arkesteijn, M. H., Koutamanis, A., & den Heijer, A. C. (2020). Towards a smart campus: Supporting campus decisions with internet of things applications. *Building Research & Information*, *49*(1), 1–20. https://doi.org/10.1080/09613218.2020.1784702