Feature Selection Mechanism for Attention Classification using Gaze Tracking Data

Ahsan Raza Khan, Syed Mohsin Bokhari, Sara Khosravi, Sajjad Hussain, Rami Ghannam, Muhammad Ali Imran, and Ahmed Zoha*
James Watt School of Engineering, University of Glasgow, United Kingdom
Email: {a.khan.9, s.khosravi.1}@research.gla.ac.uk,
{Sajjad.Hussain, Rami.Ghannam, Ahmed.Zoha, Muhammad.Imran}@glasgow.ac.uk,
smohsin599@gmail.com, *Corresponding Author: Ahmed.Zoha@glasgow.ac.uk

***Abstract*—The Covid-19 outbreak has caused disruptions in the education sector, making remote education the dominant mode for lecture delivery. The lack of visual feedback and physical interaction makes it very hard for teachers to measure the engagement level of students during lectures. This paper proposes a time-bounded window operation to extract statistical features from raw gaze data, captured in a remote teaching experiment and link them with the student's attention level. Feature selection or dimensionality reduction is performed to reduce the convergence time and overcome the problem of over-fitting. Recursive feature elimination (RFE) and SelectFromModel (SFM) are used with different machine learning (ML) algorithms, and a subset of optimal feature space is obtained based on the feature scores. The model trained using the optimal feature subset showed significant improvement in accuracy and computational complexity. For instance, a support vector classifier (SVC) led 2.39% improvement in accuracy along with approximately 66% reduction in convergence time.**

***Index Terms*—Attention classification, gaze tracking, mind wandering, feature engineering, attention monitoring.**

I. Introduction

It is common to experience distractions or attention diversion, where the individual's focus is diverted from the related task. There is a significant impact of unintended drift on learning behaviour and performance, particularly for the students [1]. The traditional class provides visual feedback for teachers, allowing them to track student learning behaviour and intervene when necessary [2]. The outbreak of the Covid-19 pandemic has disrupted the education of millions of students around the globe. The academic institutes were forced to shift toward remote teaching and learning models for delivering lectures. This model lacks the visual feedback for teachers and tools to monitor the students' attention levels continuously during online teaching sessions [3]. Therefore, it is challenging to assess the student's concentration level during online lectures. The additional challenge associated with remote learning is the distraction or attention diversion due to the surrounding environment. Furthermore, the students are more prone to distractions due to the use of digital media and smartphones during online lectures [3].

Various wearable sensor-based solutions for attention monitoring are proposed in the literature to overcome the challenge of attention monitoring in remote learning. However, gaze or eye-tracking is a non-intrusive technique to monitor the attention level of individuals due to the so-called eye link, which is highly related to visual attention. Many psychological studies use gaze-tracking to predicate the relationship between eye movements and attention level, providing the valuable insights in the context of consumer preference during online shopping [4]. Furthermore, it is also used for emotion processing, and psychopathology, including autism and attention deficit hyperactivity disorder (ADHD) [5].

The use of gaze-tracking is also gaining popularity in academia to monitor the attention level and mind-wandering using visual stimuli in remote learning session [3], [6]. For instance, the study in [7], exploited the gaze features to classify students' engagement levels in a self-placed reading activity. For raw data collection, commercial eye-tracker device is used. Furthermore, thought probes were used to report the mind wandering during the reading activity. The study in [8], provides the evidence of using gaze-tracking data to classify the attention level of individuals while engaging to a visual stimuli. The participants watched a short video as a stimuli without any distractions and also asked to perform the mental calculations while engaging, achieving an overall accuracy of 80.6%.

However, most of the available solutions use commercial grade devices which incurs an additional capital cost, limiting the scalability at a massive level. Therefore, this work aims to develop a scalable eye tracking platform which exploits the ordinary webcam to capture and analyse the gaze data. The idea is to extract statistical features for raw gaze data and link them with the attention level of students in a remote learning. Furthermore, we propose a time-bounded window operation to extract the statistical features and performed feature processing using recursive feature elimination (RFE) and SelectFromModel (SFM). The combination of feature selection techniques provides the optimal subset of features to overcome the problem of over-fitting and significantly improve the performance of the machine learning (ML) model used for attention classification. The main contributions of this work are:

- Statistical feature extraction for raw gaze data based on the time-bounded operation to link with the attention level of students.
- Proposed a feature selection mechanism using the RFE

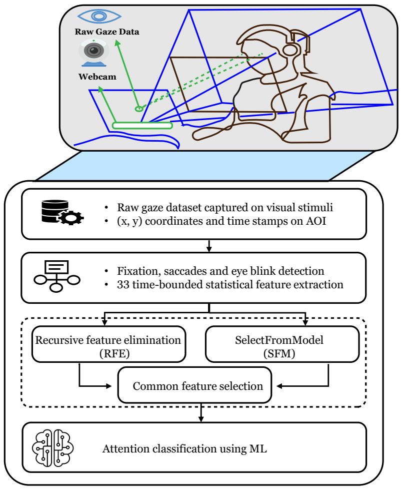


Fig. 1: The architecture of proposed gaze-based attention classification system.

and SFM to obtain the optimal subset of gaze features to overcome the problem of over-fitting for attention classification.

The rest of the paper is organized as follows: Section II covers the proposed methodology. Section III presents the results and discussion while Section IV concludes the study.

II. Proposed Methodology

This work aims to develop privacy-aware and low latency attention classification system, leveraging eye-tracking data captured by a standard webcam. A web application is developed using the open-source eye tracking library webgazer.js for data collection [9]. This platform is named as Exploring Eye-tracking Data to Support E-learning (EXECUTE) which runs on the end user browser to ensure privacy and perform local computations. The system level architecture of EXECUTE interface that captures the 2D raw gaze coordinates and timestamps as shown in Fig 1.

Twenty-five volunteers from our research group participated in the experiment and a short video lecture is used as stimuli for raw data collection. The video lecture used as stimuli consists of multiple slides with each slides having pre-defined areas of interest (AOI) to streamline the data processing and feature extraction. A low-tone acoustic bell and a feedback button are used to capture distractions in a self-caught manner. The bell rang at regular intervals, and participants had to press the feedback button to report distractions. The details of EXECUTE interface and data collection process are given in our previous study [10]. This work proposes a feature engineering mechanism for attention classification of students in a remote learning experiment using gaze metric fixations and saccades with the time-bounded operation. After regress analysis, the best features are used to train machine learning (ML) for attention classification. The results and computational complexity are compared with models trained without feature selection.

A. Data Processing and Feature Engineering

The raw data collected in this process only contains the 2D (x,y) coordinates along with the time stamps. Using an open-source tool pyGaze, the 2D raw gaze coordinates are converted into eye movements based on a dispersion-velocity filter [11]. The commonly used gaze metrics i.e., eye-blink, saccades, and fixation, are extracted from raw data. Fixation is one of the widely used gaze tracking features, which is the minimum duration where a person is trying to focus at a particular point during the visual engagement. Saccade has a variety of definitions; however, the commonly accepted one is the time duration between two consecutive fixation points. In the case of eye-blink, the loss of data by eye tracker for a short period range from 83 to 400 ms. In the initial data cleaning process, eye blinks and other missing data are removed, and statistical feature extraction is performed using fixations and saccades.

1) Feature Extraction: Once filtered data is obtained, different statistical features are extracted using saccades, fixations, AOI, and fixation scan path. The video lecture used as stimuli has pre-defined AOI per slide. A time window operation is performed to obtain the 33 statistical features in relation to saccades, fixations and scan-path on each AOI. Some features extracted from raw data include number of fixations in each AOI, percentage fixation duration in AOI, largest fixation duration, mean fixation distance, dispersion spread of two consecutive fixations, and maximum duration of fixation per AOI. The lecture content was arranged in particular order per slide, ensuring the pre-defined starting and end time for each AOI. After that, the student attention is linked with the statistical feature using the video timestamps and screen location coordinates on each AOI to obtain the attention score matrix, separating the participants into two classes i.e. attentive and non-attentive. The details on data labelling process and obtaining the attention score matrix are discussed in our previous study [10].

2) Feature Selection: Feature selection is a technique used for dimensional reduction by choosing the most suitable feature vector used for training process. The redundancy in features causes over-fitting, which degrades the model performance. Various feature selection techniques like the variance threshold method, univariate feature selection, sequential approach, feature selection as a part of pipelining, SFM, and RFE are available in literature [12]. Before applying any feature selection technique, the variance of each feature vector is calculated and all low variance feature vector are dropped. This results in reduction of feature space from 33 to 28. To further reduce the feature space, this work adopts the the combination of RFE and SFM to obtain the optimal features for attention classification. The details on both techniques are given below:

a) Recursive Feature Elimination RFE is a feature selection which utilises the ML model to eliminate the least important features after recursive training. In RFE, the estimator or ML is trained with an initial subset of features, and *coef_* attribute is used to obtain the feature importance. In the

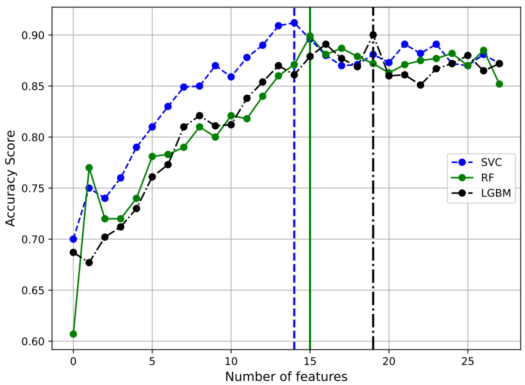


Fig. 2: Recursive features elimination using cross-validation.

process, the least important features are pruned for the current feature set and repeated until the desired number of features are reached. This work used a support vector classifier (SVC), random forest (RF), and light gradient boosting machine (LGBM) models for recursive training. The 5-Fold cross-validation is done to obtain the optimal number of features, and the accuracy score is measured using training data. Fig. 2 shows the results of the accuracy score vs the optimal number of features for SVC, RF, and LGBM.

b) Feature Importance SFM is a meta-transformer technique which can be used with any ML model to assign importance to each feature using the *coef_*attribute. This approach considers all features at once and provides a numerical score based on the contribution of each feature in the training process. For SVC, the L1 norm penalises unimportant features and eliminates them. The other technique used for feature selection is RF and LGBM which are based on the mean decrease in impurity (MDI) or Gini Importance (GI). For a given feature, this approach measures the total reduction of loss on all splits and ranks the feature according to the GI [12]. The feature with lowest loss is ranked at the highest level.

B. Model Training for Attention Classification

This work treated the attention monitoring as a binary supervised classification problem with two classes i.e. attentive and non-attentive. Before feature selection and model training, multiple-time bounded features were extracted, and discussion on them is beyond the scope of this paper. Once the feature engineering is done, the next step is selecting a suitable classifier that provides a better generalisation for unseen data with low convergence and inference time.

This is a pilot study which involves only 25 participants, and the data collected during the process is very small. Therefore, we have to consider a simple model that considers the bias-variance trade-off and the low computational complexity. For this problem, we used logistic regression (LR), decision trees (DT), RF, SVC, and LGBM and performed an extensive comparative analysis. Grid search is done to obtain the optimal hyper-parameters for each classifier. Once feature engineering is completed, each technique ranks the feature in ascending order based on importance. To get the best possible feature combination for model training, this paper combines the top 12 features obtained by RFE and SFM using different techniques ML techniques. This approach eliminates the reliance of the ML model on specific feature selection techniques and improves the overall performance.

C. Performance Metrics

In this study, the performance of the system is evaluated using accuracy, precision, recall and F-1 score. These metrics are represented mathematically as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

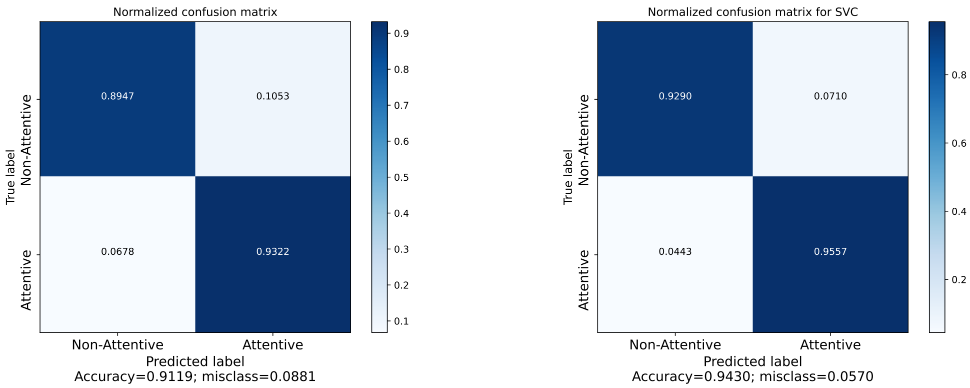
$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

where TN stands for true negative, TP true positive, FN false negative, and FP false positive.

III. Results and Discussion

In this study, attention monitoring is treated as supervised ML problem where labeled data is used for model training. To obtain an optimal classifier for attention classification, regress feature engineering is done, and the 12 best features are selected, which were obtained for RFE and SFM using different ML models. LR, DT, RF, SVC, and LGBM are the five classifiers trained using the selected features. Each classifier is trained with and without feature selection, and results are compared to evaluate the performance. The results of each classifier without feature selection are given in Tab. I. The results show that SVC has the highest accuracy of 91.91% with a convergence time of 0.06s. However, LR has the lowest convergence time, but the accuracy is a bit lower compared to SVC. The Tab. II gives the results of each classifier after feature selection. With feature selection, LGBM and SVC have the highest accuracy of 94.01% and 94.30%, respectively. It is evident from the results that feature selection has improved the overall performance of the ML model. Furthermore, with dimensional reduction, the convergence time is also significantly improved. For instance, in SVC, the convergence time is reduced from 0.06 to 0.02s, respectively. These results were expected as feature selection removes the redundant features, which caused over-fitting. Furthermore, with a small number of features, the ML model requires less time for convergence. Fig. 3 shows the confusion matrix of the SVC classifier before and after feature selection. The confusion matrix can also be used to obtain the other performance metrics like precision, recall and F-1 score. We believe that the performance of the model can be improved using more complex ML techniques. However, this will require more data and higher convergence time, which might limit the system's scalability in real-time deployment.



(a) Confusion matrix for SVC before feature selection (b) Confusion matrix for SVC after feature selection

Fig. 3: Confusion matrix for top classifier before and after feature engineering.

TABLE I: Classification results before feature selection.

Model	Accuracy	Precision	Recall	F1-Score	Convergence Time
DT	87.31	0.869	0.871	0.87	0.026
LR	88.86	0.89	0.893	0.892	0.024
LGBM	88.34	0.89	0.88	0.88	0.051
RF	90.67	0.906	0.91	0.91	0.141
SVC	**91.91**	**0.909**	**0.91**	**0.906**	**0.060**

TABLE II: Classification results after feature selection.

Model	Accuracy	Precision	Recall	F1-Score	Convergence Time
DT	88.06	0.88	0.88	0.88	0.008
LR	91.05	0.91	0.91	0.91	0.002
RF	92.75	0.92	0.92	0.92	0.092
LGBM	94.01	0.939	0.94	0.94	0.032
SVC	**94.30**	**0.94**	**0.94**	**0.94**	**0.020**

IV. Conclusions

This paper proposed a feature selection mechanism for attention classification using gaze-tracking data in a remote teaching session. Each slide in the lecture has pre-defined AOI, and a time-bounded operation is performed on each AOI to obtain 33 statistical features linked with the students' attention. Regress feature engineering is performed to remove the redundant features to deal with the problem of over-fitting. RFE and SFM are used with different ML models to obtain the best feature scoring for each technique. The best features from each model are combined to train the LR, DT, RF, SVC and LGBM, and results are compared without feature selection. The results show significant improvements in terms of accuracy and reducing the convergence time. For instance, the accuracy of SVC without feature selection was 91.91%, which improved to 94.30% after feature selection. Furthermore, it also reduced the convergence time from 0.060 to 0.020s. This study shows that the optimal feature selection improves the model performance and can easily be deployed in real-time studies. This study can be extended to link the gaze data with the psychological and emotional state of a person for the early detection of anxiety and depression.

References

[1] A. Vehlen, I. Spenthof, D. Tönsing, M. Heinrichs, and G. Domes, "Evaluation of an eye tracking setup for studying visual attention in face-to-face conversations," *Scientific reports*, vol. 11, no. 1, pp. 1–16, 2021.

[2] A. F. Abate, L. Cascone, M. Nappi, F. Narducci, and I. Passero, "Attention monitoring for synchronous distance learning," *Future Generation Computer Systems*, vol. 125, pp. 774–784, 2021.

[3] K. K. Loh, B. Z. H. Tan, and S. W. H. Lim, "Media multitasking predicts video-recorded lecture learning performance through mind wandering tendencies," *Computers in Human Behavior*, vol. 63, pp. 943–947, 2016.

[4] W. Wong, M. Bartels, and N. Chrobot, "Practical eye tracking of the ecommerce website user experience," in *International Conference on Universal Access in Human-Computer Interaction*. Springer, 2014, pp. 109–118.

[5] D. Rojas-Líbano, G. Wainstein, X. Carrasco, F. Aboitiz, N. Crossley, and T. Ossandón, "A pupil size, eye-tracking and neuropsychological dataset from adhd children during a cognitive task," *Scientific data*, vol. 6, no. 1, pp. 1–6, 2019.

[6] J. Madsen, S. U. Julio, P. J. Gucik, R. Steinberg, and L. C. Parra, "Synchronized eye movements predict test scores in online video education," *Proceedings of the National Academy of Sciences*, vol. 118, no. 5, 2021.

[7] K. Rayner, "Eye movements in reading and information processing: 20 years of research." *Psychological bulletin*, vol. 124, no. 3, p. 372, 1998.

[8] S. D'Mello, J. Cobian, and M. Hunter, "Automatic gaze-based detection of mind wandering during reading," in *Educational Data Mining 2013*, 2013.

[9] A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang, and J. Hays, "Webgazer: Scalable webcam eye tracking using user interactions," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*. AAAI, 2016, pp. 3839–3845.

[10] A. R. Khan, S. Khosravi, S. Hussain, R. Ghannam, A. Zoha, and M. A. Imran, "Execute: Exploring eye tracking to support e-learning," in *2022 IEEE Global Engineering Education Conference (EDUCON)*, 2022, pp. 670–676.

[11] E. S. Dalmaijer, S. Mathôt, and S. Van der Stigchel, "Pygaze: An open-source, cross-platform toolbox for minimal-effort programming of eyetracking experiments," *Behavior research methods*, vol. 46, no. 4, pp. 913–921, 2014.

[12] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.