

Process Eye-Tracking Data with Machine Learning Approach in Context of Fundamentals of Electrical Engineering

Johannes Paehr, Thomas Norbert Jambor*

*Faculty of Electrical Engineering and Computer Science,
Leibniz University Hannover, Germany*

**paehr@dei.uni-hannover.de*

Article history

Received

28 November 2025

Received in revised form

12 December 2025

Accepted

12 December 2025

Published online

27 December 2025

Abstract

In this research study, first-semester students solved tasks in the context of the fundamentals of electrical engineering. They were wearing eye-trackers while solving the tasks. The collected data is used to train a machine learning model per task, that predicts based just on eye-tracking data, if a student is about to succeed in solving a specific task or if the student is about to fail. The trained models reach an accuracy of 85% respectively 91% depending on the task. In the future, this model will be integrated into a virtual environment where eye-trackers are present, to assist those students who might fail.

Keywords: Electrical Fundamentals, Eye-tracking, Machine Learning

Introduction

Electrical engineering students in their first semester tend to have difficulties with the fundamentals of Electrical Engineering. This is due to several reasons. Almost half of the students in the first semester have a different first language than German. So for some of these students, the language might be a barrier. Additionally, in some countries, the direction of electric current is taught differently because the charge carriers of the opposite polarity are used. Although this is not a problem from a technical point of view, it can lead to comprehension problems that represent a learning barrier. In particular, functioning mental models are not questioned and can still be an obstacle to learning when building up new knowledge. These not-questioned mental models are called preconceptions because they might be generally valid, but they also might be wrong.

Furthermore, there are different ways to obtain the university entrance qualification. Besides the possibility of obtaining the qualification in a foreign country, there are several options to obtain the qualification in Germany. All of them confirm their university entrance qualification, but the levels of knowledge in mathematics and physics are different.

This means that the students, especially in the first semesters, build a group of learners with several different preconceptions, some of them true, but not all of them. For this reason, analysis tools such as eye-trackers are particularly interesting because they provide insight into students' behavioral patterns.

In the following, we will describe how we collected eye-tracking data and how we used it to train two machine learning models that can predict whether a student will solve a task correctly or not. In the third

step, the link to the learners is established. Here, a decision can be made on how to handle the results of the model. The first two steps are part of this article. The third step will be discussed in the Application section.

Related Work

The gaze behavior of the eyes reveals several insights into the current cognitive states of humans (Schindler & Lilienthal, 2019). Therefore, Raptis et al. highlight in their literature review the possibility of recognizing patterns in the gaze behavior, which can be recorded using eye-trackers (Raptis et al., 2016). The authors describe findings about the cognitive differences in visual perception tasks, which can be observed in eye-tracking data.

These observations were confirmed by Singh et al. when they showed that it is possible to train a machine learning model that can identify reading patterns that are typical for inspectors who should find common fault types in requirement engineering. The dataset contains eye-tracking recordings of 39 participants. As classifiers, Singh et al. used Bayesian, SVM, Ensemble, Tree, and Lazy Learners. (Singh et al., 2018)

Pritalia et al. propose an approach that detects the learning style (process information) of students based on eye-tracking data (Pritalia et al., 2020). They tested three different machine learning models (SVM, Naive Bayes, and Logistic Regression) and achieved an accuracy of 71% with the best approach. The dataset used in this study comprises 68 participants.

Many other papers have been published dealing with eye-tracking, but to the author's knowledge, none deal with preconceptions in the fundamentals of electrical engineering. However, all these publications

show that machine learning models are suitable for extracting information from this type of data.

Study

Design of the Course

To help all of the students reach an almost equal level of knowledge in the fundamentals of electronics, we offer a course that is specially designed to detect and reduce misconceptions. A lot of the tasks the students have to solve are taken from Kautz (Kautz, 2010). The focus of the course is on developing helpful conceptions, respectively, to correct unhelpful conceptions about the physical quantities of current, voltage, and resistance.

To achieve this, the course includes both theoretical and practical parts. First, the theory is explained by a lecturer, and in the second part, the students become active and work on tasks related to the previously explained (Jambor, 2024). The course takes place in a lecture hall.

Idea

In the future, based on the eye movements while the students are working on a task, a virtual assistant should give hints if it detects a disorganized or unsystematic approach to solve the task. To achieve such a system, it needs to distinguish between a systematic and an unsystematic approach in solving the tasks. So the challenge is to find significant metrics based on eye-tracking data to distinguish between the different approaches of the students. Each metric has its advantages and disadvantages, and whether it is appropriate for the distinction must first be examined. Once suitable metrics are found, they can be used to train a machine learning network, which can then be used to classify the approaches of students just by using eye-tracking data. The steps required to achieve the model are condensed in Figure 1. The first step is the eye-tracking survey. To get access to the data, the software of the eye-tracker manufacturer is used. Within this software, it is possible to export the collected data. Afterward, various Python scripts can be used to filter the data or perform calculations on the data. Additionally, the training and testing of the models are realized using Python scripts.

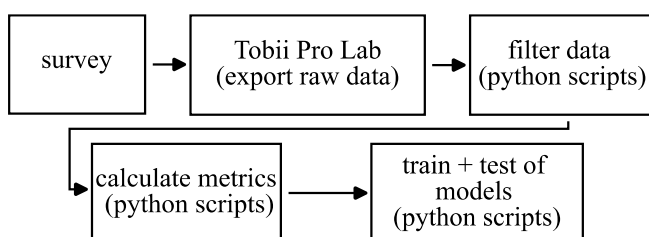


Figure 1. Steps from gathering data to the model.

Eye-Tracking Recordings

The eye-tracking recordings were made with mobile eye-trackers (Tobii Pro Glasses 2). They allow for gathering data in a realistic learning setting so that after a short time of familiarization, students behave like they would in a setting without the eye-trackers. Students have reported that they have forgotten to wear an eye-tracker. For those students who wear regular glasses, special prescription lenses were mounted to the eye-tracker because it is not possible to wear the eye-tracker and regular glasses at the same time. Due to the age of the young participants, most of them do not have aids in the form of glasses, or the prescription lenses are so low that most of the students chose not to wear the prescription lenses from the eye-tracker.

A total of 75 datasets were recorded in the course mentioned above. The datasets include 6 from female students and 69 from male students. The recordings were made in groups of six in the laboratory (the lecture hall is inappropriate for eye-tracking recordings). During the recording, the students worked individually. None of the students had to participate in the study, as it was voluntary. The students who participated in the study signed a declaration of consent, explaining who received the data and what may be done with this data. Students who took part in the study had no advantages or disadvantages compared to other students.

There are 48 usable recordings. 27 recordings are not usable because:

- The students looked under the glasses instead of through them, which makes it difficult for the eye-tracker to track the eyes at all. This happened especially to students who were not familiar with wearing glasses.
- The students started the task but did not finish it.
- The glasses slipped out of place. This can be observed on a heatmap that visualizes the fixation density when empty areas have a high density of fixations. Still, if the shift is small, the data might be usable, or if it is just a single shift, the data might be correctable. For this study, the shifted data is not taken into account.

Not all the tasks in the course are suitable for eye-tracking recording. Especially the practical parts where students have to place components on a breadboard and do some electrical voltage and current measurements are difficult to analyze because of the constantly changing surroundings. The latter results in almost no support for software-assisted analysis, so practical tasks are not recorded.

A task that is well-suited for recording with an eye-tracker is shown in Figure 2. The picture shows five electrical subnetworks. Three of them contain light bulbs, and the other two consist just of an open circuit or an ideal conductor. The first subtask for the students is to arrange the networks in ascending order (by their resistance).

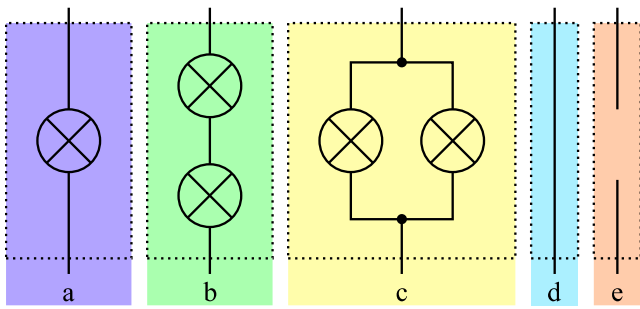


Figure 2. Students have to sort the electrical subnetworks by their resistance. The colors represent the areas of interest (AOIs).

To analyze the eye movements in such a task, areas of interest (AOIs) are defined (Holmqvist et al., 2015). Each AOI is colored in a different color. These colors are not printed on the paper the students are working on, they are just a visualization for the researcher. With defined AOIs, it is possible to generate several metrics that provide insights into students' behavior. The metrics used in this paper are:

- Dwell Strings (DS), which add an AOI-specific identifier to a chain of identifiers each time the student visits the AOI. The characters of the AOI itself are used as identifiers. If an identifier appears multiple times in a row without another identifier in between, the repetitions are removed from the DS. The reason for this is the comparability of different DSs. The DS should not map the processing speed of the tasks the students solve. Therefore, repetitions within the DS are removed.
- Dwell Time (DT_i) represents the duration of the dwell. The DT_i is used to compensate for the removal of the repetitions in the DS.
- Revisit Count (RC) represents the number of revisits in an AOI within a DS.

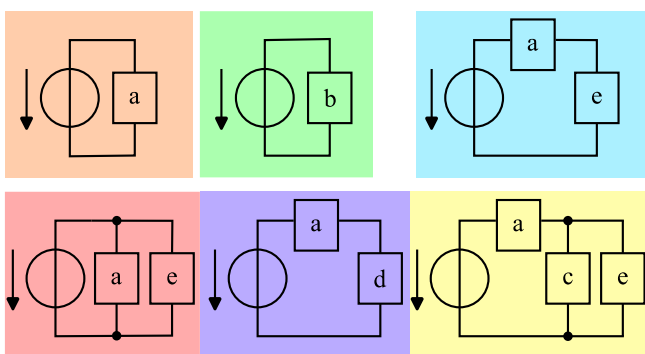


Figure 3. Students have to insert the subnetworks from Figure 2 and sort the circuits in ascending order by current. The colors represent the areas of interest (AOIs).

In the second task (see Figure 3), the students have to sort the circuits by the amount of current flowing. To do this, they have to fill in the boxes with the characters by the subnetworks in Figure 2. A typical difficulty in the fundamentals of electrical engineering is the effect

of an electrical consumer in a network. While students typically understand that an electrical consumer acts as a barrier to current, the interaction of electrical consumers connected in series or parallel is a challenge. The tasks in this article address precisely this problem.

The core idea behind the tasks is that the physical consequences in series and parallel connections can be estimated based on the electrical current. Calculations are deliberately omitted at this stage, as the aim is for students to gain not only a mathematical understanding of electrical consumers, but also an idea of how they work within a circuit. The second task in particular practises this by requiring the electrical currents of the subcircuits to be compared with each other in different configurations.

Selection of Appropriate Metrics

Initially, it is not known which eye-tracking metrics might be suitable to distinguish between a student whose approach is systematic and a student whose approach is unsystematic. This is why several metrics are calculated, and those that are significant to differentiate between systematic and unsystematic approaches are used.

None of the metrics is normally distributed, which is why rank-based tests are used to measure the significance. Both the U-Test as well as the Kruskal-Wallis test indicate that the RC is within the significance level of $P \leq 0.05$. For the first task, the Kruskal-Wallis test calculates a p-value of 0.037 for AOI E. The RC of the other AOIs of the first task does not seem to be relevant, because the p-value is bigger than the significance level. For the second task, the p-value of the RC reaches the significance level with the top three AOIs in Figure 3 (0.021, 0.010, 0.010).

Machine Learning Model

Motivation

In previous work, other tasks from this course were analyzed and used to train different machine learning-based models (Paehr & Jambor, 2023). The best performance was achieved with a Long Short-Term Memory-based (LSTM) approach. The data used for training and testing the model was exclusively based on DS. In terms of the target application, in which students need to be identified by the application as not following a systematic approach to receive support, this is a problem. The entire DS is present when the task is completed by the student or when the student gives up, respectively, solves the task wrong without noticing the inconsistencies.

If the students are to receive hints, then they must receive them earlier, during processing. Based on the DS, this is not possible because the DS is only available after the task is completed. A second problem of this approach lies in the different lengths of the DSs (see

Figure 4). Each student generates a different DS with their gaze. As mentioned earlier, the students may have highly different preconceptions, so the lengths of the DSs differ. Unhelpful conceptions can lead to patterns in the DS that are not helpful. Consequently, it is likely that unhelpful conceptions will lead to incorrect sorting. However, the model does not allow us to draw conclusions about specific conceptions. Nevertheless, lecturers can benefit from knowing that problems frequently arise with specific tasks. The shortest DS is 15 characters long, while the longest DS is 287 characters long. These differences impede the ability of typical comparison algorithms, such as the Levenshtein distance (Levenshtein, 1966), to compare the DSs of the students with each other, because the length may have a bigger influence than the content of the string itself. The Levenshtein distance penalizes missing characters as much as not matching characters. This means that if the DSs to be compared are not nearly equal in length, it is not appropriate to use this algorithm to compare the DSs.

A similar problem exists with LSTM models because the input data of such a model needs to be equal in size. Consequently, the longest DS serves as the basis for determining the input size of the LSTM model. Therefore, the shorter DSs are filled up with zeros to match the size of the longest DS.

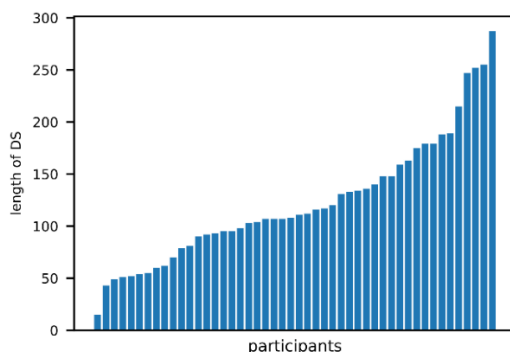


Figure 4. Different lengths of DSs in the first task.

This is why the DS has been divided into chunks (see Figure 5). The chunks consist of shorter parts of the entire DS. Consequently, it is expected that the prediction of a trained model is not as accurate as with the entire DS, but it opens up the possibility of getting the prediction earlier. The first prediction can be obtained after the student has reached the number of transitions between AOIs necessary to build the first chunk of the minimal length.

DS:	A	B	A	C	D	A	E	A	E	...
chunk ₁ :	A	B	A	C	D					
chunk ₂ :		B	A	C	D	A				
chunk ₃ :			A	C	D	A	E			
chunk ₄ :				C	D	A	E	A		
chunk ₅ :					D	A	E	A	E	

Figure 5. DS divided into overlapping chunks. In this example, the length of the chunks is 5.

So the research questions are as follows:

- RQ1: How many transitions are necessary to create a model that can predict the outcome of the student's work with sufficient accuracy?
- RQ2: How many parameters should such a model have to minimize overfitting and underfitting?
- RQ3: Can other significant metrics improve the performance of the model?

Design of the Model

In (Paehr & Jambor, 2023), eye-tracking data from a comparable task were analyzed with different machine learning models, like the decision tree (DT), the support vector machine (SVM) (Schölkopf & Smola, 2002), the hidden Markov model (HMM), and also a neural network with a bidirectional LSTM layer (BiLSTM-Net). The DT and the SVM were trained with position-based measures such as the fixation duration on an AOI. While these kinds of metrics are significant for this task, the results demonstrated that these models achieve an accuracy of 0.79 respectively 0.73 in predicting the success of students on a specific task. Additionally, sequence-based approaches (HMM and BiLSTM-Net trained with DS) were evaluated. The HMM achieved an accuracy of 0.73 while the BiLSTM-Net reached an accuracy of 0.8. The accuracy of the BiLSTM-Net is the reason why this model is adapted to the tasks here.

The base architecture of the models consists of LSTM and dense layers (see Figure 6). While the LSTM is an advanced version of a recurrent neuronal network (Hochreiter & Schmidhuber, 1997), it is designed to store input information and detect sequences over time efficiently. Therefore, the LSTM layer/s is/are responsible for learning the crucial dependencies that lead to success or failure, within the DSs. On the left side, two LSTM layers, each with 10 units, are stacked. The number of units is a control variable of the possible complexity of the model. The fewer units, the less complex the model can be.

Tests with the dataset have shown that the model fits better if two LSTM layers are stacked compared to a single layer with twice the number of units. This side of the model is responsible for learning patterns within the DSs and the corresponding DTis. This is why the input shape of this side is chosen to [None, chunk size, 2]. The first parameter is a placeholder for the number of datasets. Because the number of datasets depends on the chunk size, it varies through the different chunk sizes of the different trainings (within a single training, the chunk size stays constant). The second parameter is the chunk size, and the final parameter is set to two for the AOI position and the corresponding DTi. A dropout of 10% is added to the two LSTM layers to make the model more robust.

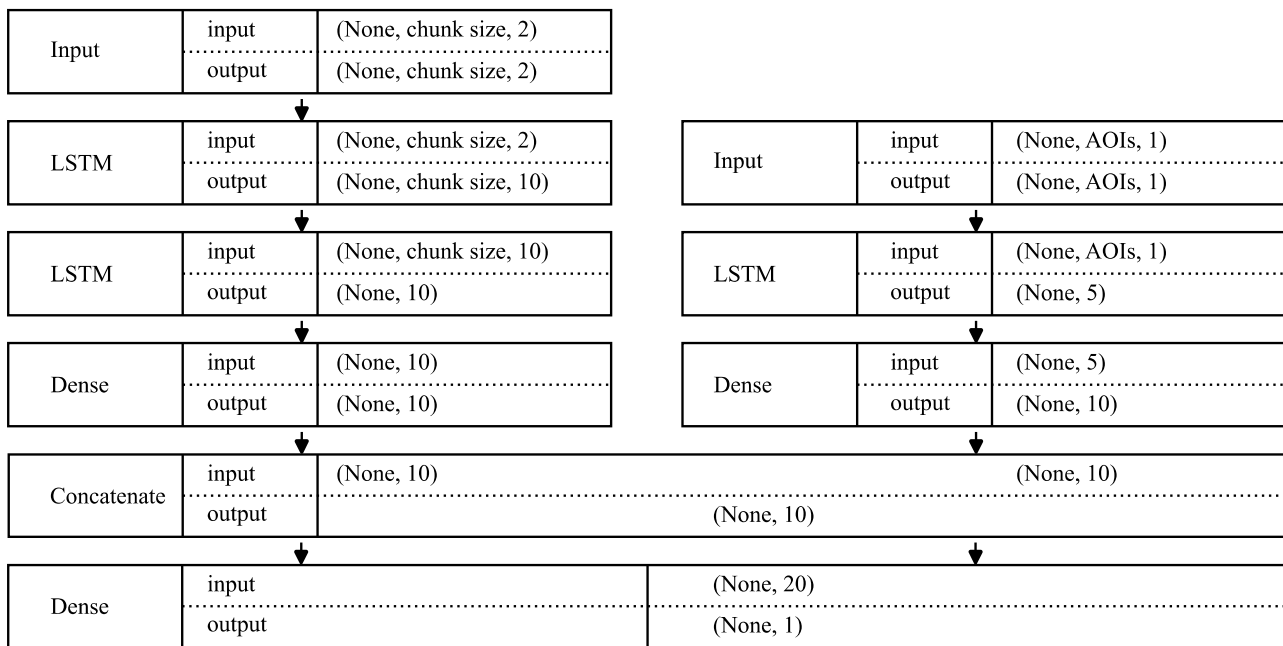


Figure 6. Architecture of the model: left: processing of the DS and the DTi; right: processing of the RC

The right side of the model consists of just one LSTM layer and a dense layer for transitioning the results of the LSTM layer to the output. This part of the model is supposed to learn information within the dataset about the RC. This is why the input shape is dependent on the number of AOIs. Consequently, the number of AOIs is five for the first task (see Figure 2) and six for the second task (see Figure 3). The number of units of this LSTM layer is comparatively small, with five units. The advantage of adding another significant metric to the model may be undermined if the number of units is chosen too large. As the results show (see Table 1 and Table 2), five units appear to be an appropriate choice.

To combine the results of both sides of the model, the concatenate layer is used. This layer does not own any weights that could be trained. It just concatenates the results of the layers above, which consist of two dense layers that are both activated by a ReLU function. The final layer of the model is a dense layer. This dense layer is necessary because the output of the LSTM layer is dependent on the number of units the layer consists of. Consequently, the dense layer is a translation layer between the outputs of the LSTM layer and the desired output of the model. For this model, just a single output is required to differentiate between the student's approaches. Consequently, a sigmoid function is used for activation. To train the model, the learning rate of the Adam optimizer is chosen to be 0.001.

Results

Due to the limited number of datasets, it is challenging to identify optimal parameter settings for the model. The number of trainable model parameters must be kept to a minimum to prevent overfitting the training data. On the one hand, this leads to poor

accuracy of the test data, and on the other hand, it might cause a positive gradient of the loss function from the test data. Both the accuracy and the loss function result in a suboptimal performance of the model on never-before-seen data (test data).

This is why our best approach of the model has two LSTM layers (each with 10 units) followed by a dense layer (also 10 units, left side). In parallel (right side), a third LSTM layer with 5 units and a second dense layer with 10 units are added to the model to allow the integration of metrics such as RC into the model (see Figure 6).

Several other combinations of architectures were tested. Increasing the number of LSTM layers also leads to an increase in the number of trainable parameters, which in turn increases the risk of overfitting. The same effect can be observed when the units of the LSTM layers are increased.

It is commonly known, that Bi-LSTM networks adapt better than LSTM networks (Schuster & Paliwal, 1997). However, this is not always the case. The bidirectional part of the network increases the trainable parameters. In this special case, the accuracy of such a model increases for the training data, but the accuracy of the test data decreases if all other settings are constant, except for the LSTM layers, which are extended to Bi-LSTM layers.

Referring to the first research question (RQ1), the impact of varying chunk sizes is evaluated. For the first task, the shortest chunk size tested is five, and the longest chunk size tested is 180. For this test, all models between these sizes are separately trained. Due to the split into different sizes, the number of datasets is bigger when the chunk size is chosen shorter (see Equation (1)).

$$k = N - c_s + 1 \quad (1)$$

N is the size of the DS, and k is the number of chunks for a chosen chunk size of c_s . Although the number of datasets is slightly higher due to this effect, the smaller the chunk size is, models trained with chunk sizes up to 30 do not adapt to the data well.

For the second task, it is appropriate to create a second model because different AOIs need to be included (see Figure 3). The basic structure of the second model is identical. The difference between the first and second model is the number of AOIs taken into account. The second task has six AOIs, while the first task just has five AOIs.

It is possible to use all AOIs from the first and the second task in one model, but this would also increase the length of the DS and create additional options for constructing the DS. Consequently, the AOIs from the first task are not used to train and test the model of the second task. The resulting lengths of the DSs from the first task are shown in Figure 7. Similar to the first task, the lengths of the DSs from the second task differ considerably.

The datasets are split into approximately 80% training data and 20% test data. The reason why it is just approximately 80% training data is that the entire chunks of one DS need to be either training data or test data. If a few chunks of one DS are designated as training data and the remaining chunks of the same DS are used as test data, it will occur that almost the same chunks are used to train and test the model. To prevent this, the training test split is done before generating the chunks. Because the DS are of different lengths, it can happen that the train test split is not perfectly 80/20.

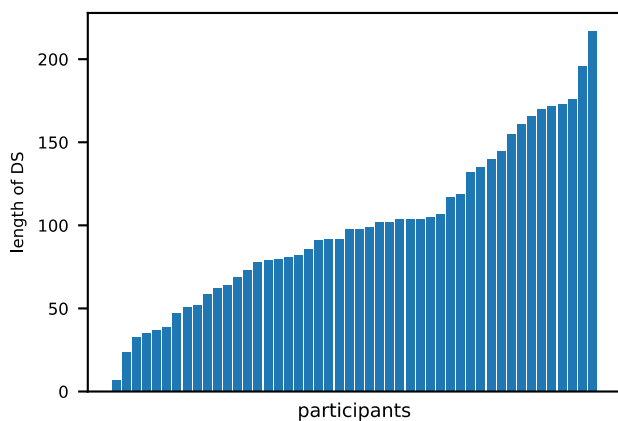


Figure 7. Different lengths of DSs in the second task.

Discussion

The number of datasets is of critical importance when working with ML approaches. Typical datasets in machine learning approaches are usually large. This is necessary because the model is supposed to generalize to fit not just data from the training, but also to never-before-seen data.

Table 1. Results of the model for the first task (see Figure 2)

chunk size	40		50		60	
	with RC	w/o RC	with RC	w/o RC	with RC	w/o RC
train	80%	-	90%	-	86%	-
test	76%	-	85%	-	81%	-

Table 2. Results of the model for the second task (see Figure 3)

chunk size	40		50		60	
	with RC	w/o RC	with RC	w/o RC	with RC	w/o RC
train	86%	83%	90%	86%	92%	-
test	85%	83%	86%	85%	91%	-

Gathering eye-tracking data is challenging. There are several reasons why most eye-tracking studies have fewer than 30 participants:

- Every eye-tracker needs to be calibrated before the recording can start. This process is not automated and consequently time-consuming.
- The eye-tracker does not work with all students. Sometimes, when the eyes of the participants are too moist, unintended reflections occur and make a recording impossible. Additionally, some types of contact lenses can be problematic.
- A problem with mobile eye-trackers is that they may slip. This is problematic when the glasses slip slightly more, as the calibration of the eye-tracker is designed to stay in position.

Nevertheless, our machine-learning approach reached an accuracy that is acceptable for the target application. The results indicate that the first model reaches an accuracy of 85% taking the metrics RC, DS, and DTi into account (see Table 1). The RC appears to be a critical factor in this model because it did not adapt well to the datasets when the RC was excluded from the analysis. Referring to RQ3, it should be noted that other significant metrics, in addition to those already mentioned, can enhance the model.

Referring to RQ1, the results demonstrate that a chunk size of 40 is sufficient to predict the result of the student with an accuracy of 76%. Considering that the majority of the students did 50 or more transitions when solving the task (see Figure 4), a chunk size in the range from 40 to 50 appears appropriate.

The second model, for the second task, performs even better (see Table 2). As observed in the previous model, the RC has a positive effect on the results in terms of test accuracy. Similar to the first task, the chunk size is sufficient to predict the success of the students before they have completed the task. A chunk size of 40 to 60 appears appropriate in this case as well.

The model size of both models is almost equal, and compared to typical machine learning approaches, fairly small. This is necessary because the model tends to overfit if the layers are chosen larger. To address this issue, more training data is necessary.

Limitations

While the models make it possible to predict whether a student will solve the tasks, further conclusions can also be drawn. For example, because the models are able to calculate this prediction, the influence of the input variables on these predictions can be determined. Since the input variables are the order in which the AOIs are gazed, the fixation lengths, and the RC, these variables are relevant for the prediction. A pattern for a systematic approach can be learned from the order in which the AOIs are gazed. The LSTM architecture used in this model is precisely suited for this purpose. Consequently, it can be assumed that the models tend to classify the systematic approach or the non-systematic approach, even though they have been trained using the results of the tasks. Since the fixation durations are also integrated into the model and there is a correlation with cognitive load (Skaramagkas et al., 2023), the model result can also be an indicator of cognitive load. These are indications of why the model arrived at the corresponding decision. However, it is not possible to say without further ado why the model arrived at the corresponding decision. It is also not possible to conclude that there is a specific preconception. There are numerous preconceptions in the fundamentals of electrical engineering, and machine learning approaches require more data sets with these preconceptions for training in order to be able to recognize the patterns of these preconceptions.

Since the tasks involve several concepts, such as parallel and series connections of electrical consumers, it is not possible to make a statement about the students' conceptual understanding of the relationships based on a binary classification of the models.

Application

There are several ways to deal with the models' predictions. On the one hand, automated support is conceivable, for example, within a VR application whose glasses have eye trackers. Depending on the model, irrelevant areas could be concealed at first. It is also conceivable that the necessary electrical rules for completing the task could be displayed optionally. If no VR environment is to be used, it is also conceivable that the model results would not directly influence the task sequence, but would only be available to the lecturer, who could repeat task-specific explanations.

The models proposed here are designed to generate the most accurate classification possible as

early as possible. This means that the decision on when to take support measures can be made from a didactic rather than a technical perspective. The appropriate timing for a measure must be examined.

The question of how model results influence teaching depends on how the results of the models are used. If subliminal optional hints are used within a VR application, the impact of an incorrect model prediction is less pronounced than if the task were to be aborted. Another possibility is that the model results are only available to the lecturer. This allows the lecturer to objectively assess which tasks pose a challenge for the students. As a result, the lecturer can focus on repeating the relevant tasks.

Conclusion

The novelty of this approach lies in the fact that two models based on an LSTM architecture have been trained using eye-tracking data from the context of electrical engineering fundamentals. Currently, the model can differentiate between students who are probably successful in solving a specific task and students who may fail. The data basis of the models is based exclusively on eye-tracking data. Consequently, conclusions can be drawn from eye movements that have not been used in a classic scenario to date, meaning that potential has not been fully exploited. Models such as those presented in this article are a step toward making this potential accessible.

To utilize the models presented here, they need to be integrated into a virtual environment where eye-trackers are present. The data from the eye-tracker must then be filtered and fed directly into the models. This allows for the generation of optional hints for students who are about to fail within the learning situation.

To generate more specific hints, a finer differentiation might be helpful. However, to achieve this with sufficient accuracy, more data is needed.

The model is trained with data from students who solve the task on conventional paper. The assumption is that this data can be used to create support in a virtual environment if the virtual environment is similar enough. If this is not the case or the behavior of the students is different in the virtual environment, the model needs to be adapted or retrained.

The fact that models can be trained exclusively using data collected by eye trackers and can achieve accuracies of up to 91% shows that a wealth of information can be gleaned from gaze directions. In this approach, locations, fixation times, and RC have been used as significant metrics. It cannot be ruled out that other significant metrics exist that would allow for a more refined classification. In this respect, any approach that provides a broader perspective on learners may be helpful.

Acknowledgement

The authors would like to thank Leibniz University Hannover for the support provided in conducting this study.

Conflict of Interest

No conflicts of interest exist.

References

- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Holmqvist, K., Nystrom, M., Andersson, R., Dewhurst, R., Jarodzka, H., & Weijer, J. V. D. (2015). *Eye Tracking A comprehensive guide to methods and measures*. A comprehensive guide to methods and measures (p. 560). Oxford University Press.
- Jambor, T. N. (2024). From Theory to Practice: Improving Learning Through Action Orientation in Academic Education. 2687–2696. <https://doi.org/10.21125/inted.2024.0741>
- Kautz, C. H. (2010). *Tutorien zur Elektrotechnik* (1st ed.). Pearson Studium.
- Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8), 707–710.
- Paehr, J., & Jambor, T. N. (2023). Using Eye-Tracking Technology to Provide Assistive Support in a Mixed Reality Learning System. 6067–6072. <https://doi.org/10.21125/iceri.2023.1516>
- Pritalia, G. L., Wibirama, S., Adji, T. B., & Kusrohmaniah, S. (2020). Classification of Learning Styles in Multimedia Learning Using Eye-Tracking and Machine Learning. 2020 FORTEL-International Conference on Electrical Engineering (FORTEL-ICEE), 145–150. <https://doi.org/10.1109/FORTEL-ICEE50915.2020.9249875>
- Raptis, G. E., Fidas, C. A., & Avouris, N. M. (2016). Using Eye Tracking to Identify Cognitive Differences: A Brief Literature Review. *Proceedings of the 20th Pan-Hellenic Conference on Informatics*, 1–6. <https://doi.org/10.1145/3003733.3003762>
- Schindler, M., & Lilienthal, A. J. (2019). Domain-specific interpretation of eye tracking data: Towards a refined use of the eye-mind hypothesis for the field of geometry. *Educational Studies in Mathematics*, 101(1), 123–139. <https://doi.org/10.1007/s10649-019-9878-z>
- Schölkopf, B., & Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press. <https://doi.org/10.7551/mitpress/4175.001.0001>
- Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), 2673–2681. <https://doi.org/10.1109/78.650093>
- Singh, M., Walia, G. S., & Goswami, A. (2018). Using Supervised Learning to Guide the Selection of Software Inspectors in Industry. *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 12–17. <https://doi.org/10.1109/ISSREW.2018.00-38>
- Skaramagkas, V., Giannakakis, G., Ktistakis, E., Manousos, D., Karatzanis, I., Tachos, N., Tripoliti, E., Marias, K., Fotiadis, D. I., & Tsiknakis, M. (2023). Review of Eye Tracking Metrics Involved in Emotional and Cognitive Processes. *IEEE Reviews in Biomedical Engineering*, 16, 260–277. <https://doi.org/10.1109/RBME.2021.3066072>