Gaze-Based Activity Recognition with a LSTM

Kasper Vaessen
Supervisor(s): Guohao Lan, Lingyu Du
EEMCS, Delft University of Technology, The Netherlands

June 19, 2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

Classification of sedentary activities using gaze tracking data can be of great use in fields such as teaching, human-computer interaction and surveilling. Conventional machine learning methods such as k-nearest neighbours, random forest and support vector machine might be used to classify such activities, but this requires knowledge about the domain to extract features. Deep learning methods such as a long-short term memory neural network do not require manual feature extraction and are therefore more accessible. To test the feasibility of using these deep learning models, this paper answers the question: Can a long short-term memory neural network (LSTM) be used for gaze-based activity recognition? It was found that a LSTM is highly suitable for user-dependent testing data with an average accuracy of 96.61%. For user-independent testing data the LSTM is less suitable with an average accuracy of 43.34%.

1 Introduction

Human activity recognition (HAR) is becoming an increasingly popular field as human sensors are used more and more. This is mainly because different domains profit from detecting human activities, such as the industrial sector, office scenarios, sport sector, entertainment sector and health care [1].

While HAR using body sensors has been around for a couple of decades, not much research has been done on HAR using gaze tracking data, even though a subject's gaze movement is highly correlated with the sedentary activity the subject is performing [2]. Detecting these sedentary activities has many applications. As described by N. Srivastava et al. [2], an example of one of these applications is time tracking software such as RescueTime[1], Time Doctor[2] and Toggl[3]. These applications keep track of what the user is spending their time on behind their computer. With this information, it either reminds them to get back to work, or it can be used to generally track their time. This is done by tracking the application which are open, but it can not check what specific task the user is doing within that application. For example, it can't detect whether the user is reading a article or simply staring at their screen. This is where classification of activities using gaze tracking can be useful. Combining these techniques can lead in more accurate detection of activities.

Detecting these activities is a typical classification problem, on which often machine learning methods are applied. In the field of HAR mostly conventional machine learning models are used, such as kNN [3, 4], random forest [5, 6] and SVM [7]. These methods work well, but require knowledge about the data to design features which the models can train and classify on. Deep learning methods such as the convolutional neural network (CNN) and long short term memory neural network (LSTM) can extract the features automatically and therefore do not require manual feature extraction. Especially the LSTM seems suitable for the purpose of HAR [8], due to its ability to learn and remember from long sequences of input data [9].

This paper aims to test whether the LSTM can be used for gaze-based activity recognition. This will be done by designing, training and optimizing the LSTM on three data set. Afterwards, this LSTM will be evaluated on each of these data sets and it will be compared to a kNN, random forest, SVM and CNN in terms of accuracy and robustness against heterogeneity among subjects.

[1] https://www.rescuetime.com/
[2] https://www.timedoctor.com/
[3] https://toggl.com/

1.1 Related Work

In the field of gaze-based HAR, mostly conventional machine learning methods have been used. One example is the work by Rivastava et al., who designed high level features for the kNN, Random Forest, and SVM using the sedentary data set this paper will also use [2]. They achieved a user-independent F1-score of 0.61, 0.73 and 0.71 on the kNN, Random Forest, and SVM respectively.

Kunze et al. have performed similar research on a reading data set, which is also used in this paper [10]. They achieved 99% user-dependent accuracy and 74% user-independent accuracy.

Lan et al. have introduced a method to model the gaze data as a graph and classify based on this graph [11]. This method results in a user-independent F1-score of approximately 0.82 on the desktop data set, which is also used in this paper.

Related work using the LSTM has been performed by Chevalier, who has used the LSTM to detect human activities using accelerometer data [12], achieving an accuracy of 91% on a user-independent test set.

Ullah has used the stacked LSTM to detect similar activities [13], resulting in an accuracy of 93% on a user-independent test set.

2 Methodology

To answer the question stated in Section 1, some steps have to be taken. These steps are data pre-processing, designing and implementing the LSTM, determining and tuning hyper parameters of the LSTM and evaluating the LSTM.

In Figure 1 an overview of the classification pipeline described in this section can be found.

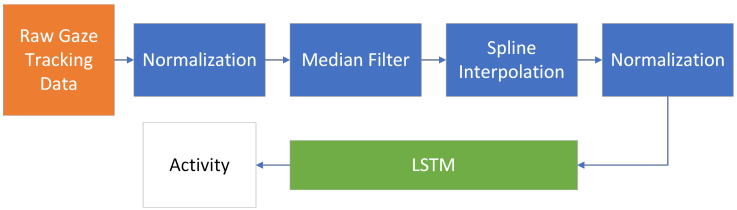


Figure 1: Pipeline overview of classification

2.1 Data Pre-processing

The raw data is not directly ready to be used by the LSTM model. This is because of three reasons: the x and y values are not in the same range, the data is noisy and there are not many time series per activity available. An example of the raw data can be found in Figure 2.

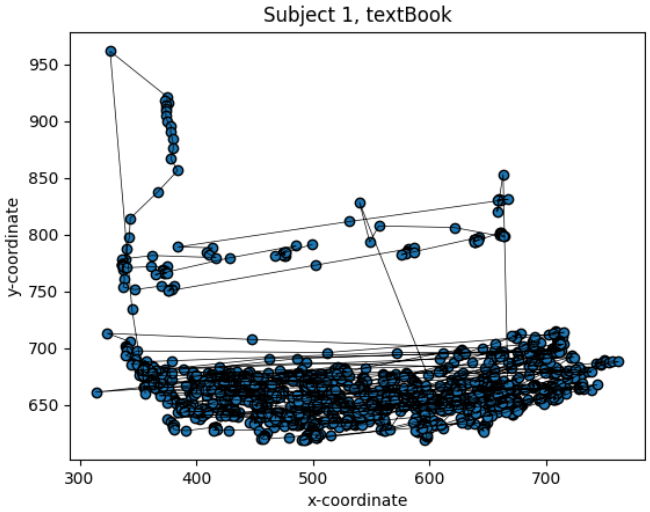


Figure 2: Raw data of Subject 1 reading a textbook (reading dataset)

The fact that the x and the y coordinate are not in the same range can result in lower accuracy scores for the model [14]. This can easily be solved by normalizing the data.

The data being noisy can be caused by multiple factors, such as the gaze tracking hardware not always being accurate, the subject being distracted from the activity or the subject blinking. These noisy data points can decrease the model performance [15]. To solve this problem, a median filter with a sliding window of 10 seconds is applied [11]. This window filters out any point which has a euclidean distance larger than a certain threshold k to the median of its sliding window. A plot of the median filter applied to the data shown in Figure 2, can be found in Figure 3. It is visible that the points which are far away from its neighbouring points are filtered out. This indicates that the data is indeed less noisy.

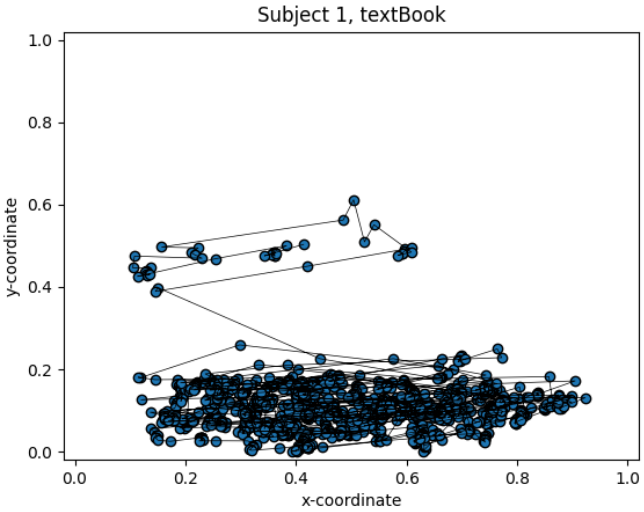


Figure 3: Normalized and median filtered data of subject 1 reading a textbook (reading dataset) with k=0.3

.

Because samples were filtered out, there is less data to train the model on, which can result in decreased performance [16]. Furthermore, a set amount of points might now no longer represent the same time interval, which would make the data inconsistent. This problem can be solved by applying spline interpolation as proposed by G. lan [11]. A plot of this interpolation can be found in Figure 4.

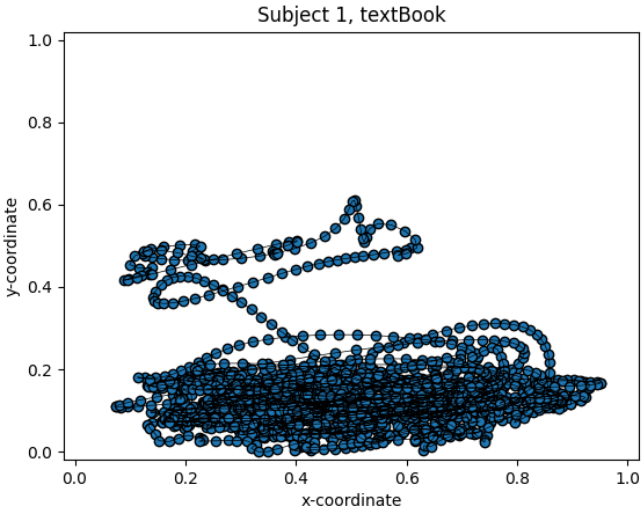


Figure 4: Normalized, median filtered and interpolated data of subject 1 reading a textbook (reading dataset) with k=0.3

.

At this point, the problem still exists that there is only one big time series of approximately 27000 datapoints per subject per activity. This is too large of a sample to classify on its own. This would also mean that there are only the amount of subjects of training samples per class. This is where the sliding window comes in. It splits the data into windows of a certain window size w with 90% overlap.

Finally, the data is normalized between 0 and 1 again to remove any variation in the x and y range.

2.2 Design and Tuning of the Model

As a basis for the LSTM, the model by Chevalier is used [12] and edited. This model includes a LSTM layer with 160 units, a dropout layer and a dense layer with 100 units. Next to this, the model proposed by G. Lan [11] is implemented and used as an comparison. The latter model excludes the dropout and dense layer which the first model has. From this baseline, the model is trained for each data set.

To further improve the accuracy, two hyper parameters were identified to be tuned: the window size w and the filter threshold k. The models are trained on all combinations for the proposed values for these parameters.

3 Results and Discussion

The LSTM described in Section 2 is evaluated on three different data sets, which are described below in Section 3.1. The results of this evaluation can be found in Section 3.2.

3.1 Data sets

The model is trained and evaluated on the following three data sets.

Reading [10]

This data set contains data of 9 subjects reading a magazine, manga, newspaper, novel, scientific paper and textbook.

Each subject read each kind of text for 15 minutes, which comes down to 27000 data points with a sample rate of 30Hz.

Sedentary Activities [2]

This data set contains data of 21 subjects performing the following activities: browsing, debugging, interpreting, playing, reading, searching, watching and writing.

Each subject performed each activity for a variable amount of time, with the amount of samples ranging from 4000 to 12000.

Desktop Activities [11]

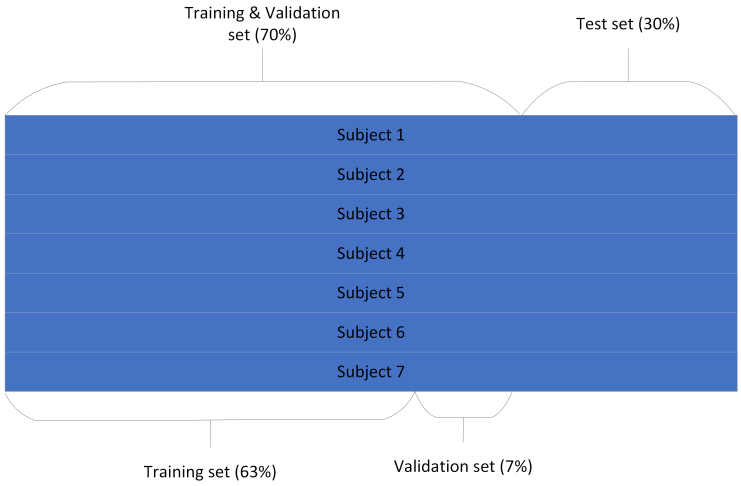
This data set contains data of 7 subjects performing the following activities: browsing, playing, reading, searching, watching and writing.

Each subject read each kind of text for 15 minutes, which comes down to 27000 data points with a sample rate of 30Hz.

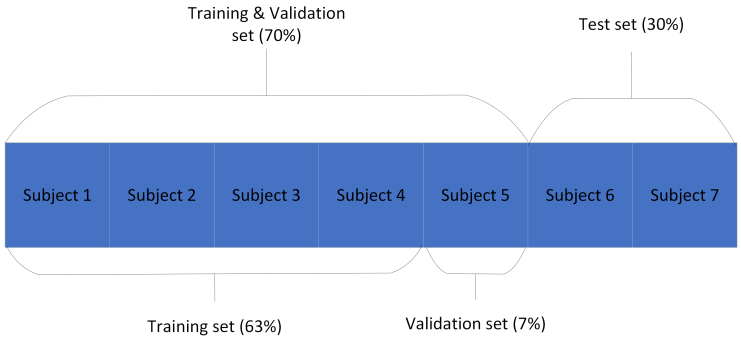
All the data sets have been split into a training, validation and test set in two different ways; namely subject-dependent and subject-independent.

If the data is split up in a subject-dependent way, this means 70% of the data from each subject is taken to be the training set, out of which 10% is taken to be the validation set. This leaves a test set of 30% to be the test set. This split is demonstrated in Figure 5a.

If the data is split up in a subject-independent way, this means 70% of the subjects (rounded to a whole number of subjects) is taken to be the training set, out of which 10% (rounded to a whole number of subjects) is taken to be the validation set and this again leaves a test set of approximately 30% of the subjects. This split can be found in Figure 5b.



(a) Subject-dependent data split



(b) Subject-independent data split

Figure 5: Division of data into training, validation and test data in a subject-dependent and subject-independent way.

3.2 Results

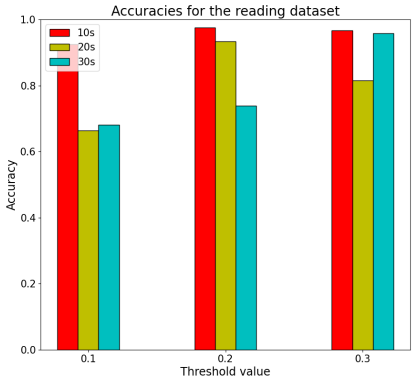
As described in Section 1 the model will be evaluated on accuracy and robustness against heterogeneity among subjects. Where accuracy is defined as follows.

Definition 3.1. Let H be the amount of correctly classified windows and T be the total amount of classified windows. The accuracy is defined as

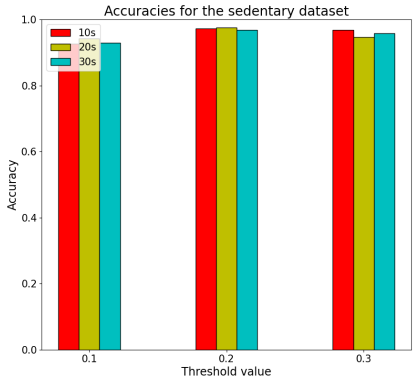
$$accuracy = \frac{H}{T}$$

In Figure 6 the subject-dependent accuracy's for all the combinations of possible assignments of the hyper parameters are shown for all three of the data sets on the model proposed by G. Lan [11], which is the model excluding the dense and dropout layer. This figure shows that on all the data sets the window size of 10s and threshold of 0.2 is performing best. The

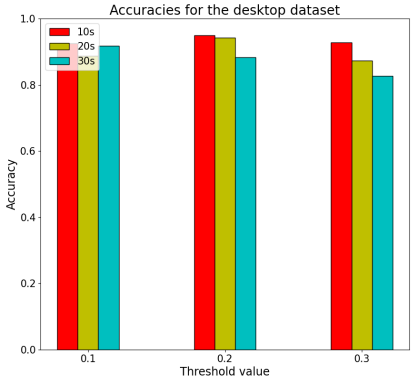
accuracy's for this combination of assigned parameters are for the reading, sedentary and desktop data set respectively 97.59%, 97.32% and 94.92%.



(a) Reading data set



(b) Sedentary activity data set

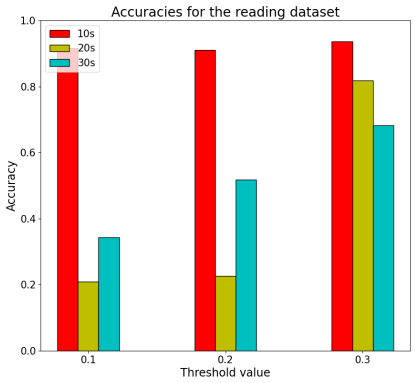


(c) Desktop activity data set

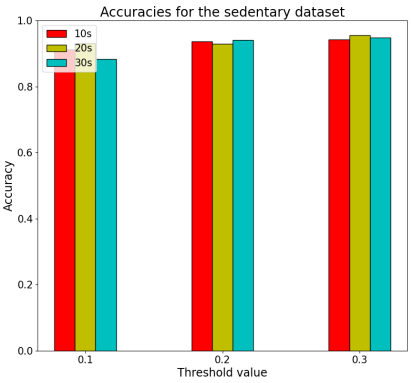
Figure 6: Accuracy's for model excluding a dense layer trained with $w \in \{10s, 20s, 30s\}$ and $k \in \{0.1, 0.2, 0.3\}$ on the three data sets.

The model proposed by Chevalier [12], of which the results can be found in Figure 7, shows worse performance than the previously proposed model without the dense and dropout layer. This is especially the case on the smaller two data sets with bigger window sizes.

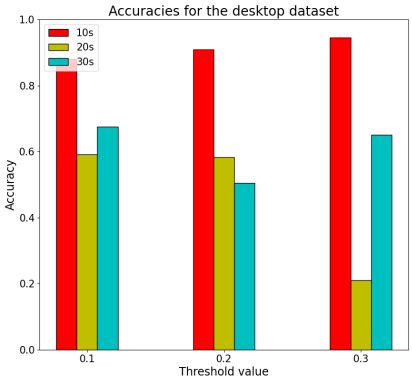
The best accuracy's for the reading, sedentary and desktop data set are respectively 93.71%, 95.51%, 94.54%. This means this model is less suited for the used data than the previously discussed model.



(a) Reading data set



(b) Sedentary activity data set



(c) Desktop activity data set

Figure 7: Accuracy's for the model including a dense layer trained with $w \in \{10s, 20s, 30s\}$ and $k \in \{0.1, 0.2, 0.3\}$ on the three data sets.

All previous results were subject-dependent as explained in Section 3.1. This means the model has learned from every subject. When training and evaluating in a user-independent way, the model has average accuracy's which can be found in Figure 8. The average accuracy of the three data sets is 43.34%. The figure shows that the model performs significantly worse on the subject-independent data when comparing to the subject-dependent data. Figure 8 also shows that the model classifies the sedentary data set significantly better than the other two data sets when splitting in a subject-independent way, especially when compared to the difference in performance of Figure 6.

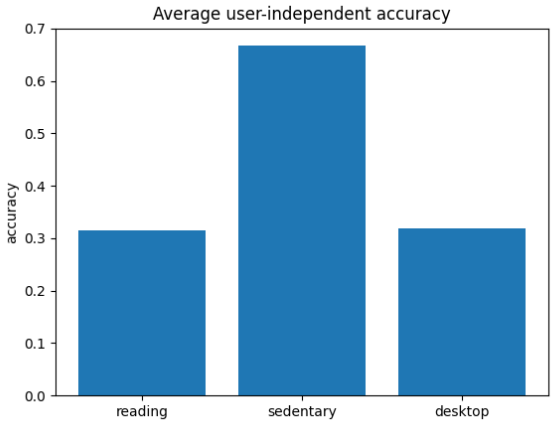


Figure 8: Subject-independent accuracy's on the three data sets with $w = 10s$ and $k = 0.2$.

Comparing the model against a CNN [17] and the more conventional machine learning models Random Forest, SVM and kNN [18–20] trained and evaluated on the same three data sets described in Section 3.1, results in the charts found in Figures 9 and 10. Here it is visible that on subject-dependent data the LSTM performs best on the sedentary data set and second to best on the other two data sets.

On the subject-independent data, the model performs worse on average than the other four models, as visible in Figure 10. On the sedentary data set however, it performs close to the best, with an accuracy difference of 2%.

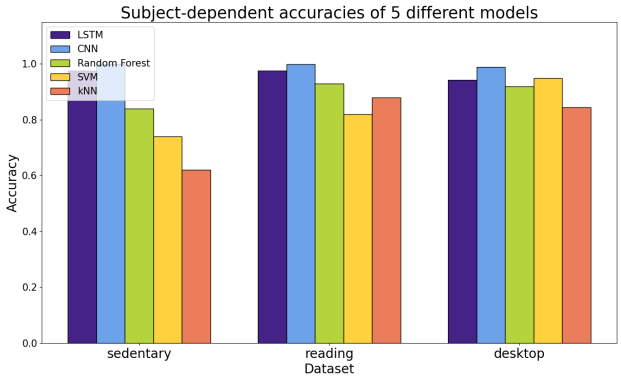


Figure 9: Subject-dependent comparison of 5 machine learning models.

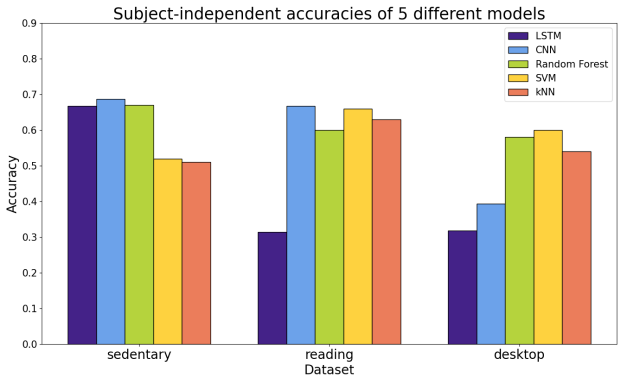


Figure 10: Subject-independent comparison of 5 machine learning models.

4 Conclusions and Future Work

4.1 Conclusions

This paper aimed to answer whether a long-short term memory neural network (LSTM) can be used for gaze-based activity recognition. This was done by designing and tuning a LSTM, after which it was evaluated on different kinds of data and compared to other models. The best performing model turned out to be a neural network with a single LSTM layer of 160 units. Besides this, a smaller window size of ten seconds for splitting the data performed best. It was found that a LSTM is highly suited when predicting with subject-dependent data with an average accuracy of 96.61%. This also on average outperforms all other models which the LSTM was compared to except for the CNN. The model is less suited when using subject-independent data, with an average accuracy of 43.34% it being the worst performing model on subject-independent data. The difference in performance between subject-dependent and subject-independent data indicates that the model is not robust against heterogeneity among different subjects.

4.2 Future Work

In the future, research could be conducted to further increase the performance of the model. Three points of improvement were identified.

The fact that the performance decreases on smaller data sets is probably because the model is overfitting to the small amount of training data [21]. This could also partially explain why the model is performing worse on the bigger window size, since a bigger window size results in less data points. Further research with a bigger data set could be conducted to test this hypothesis.

To achieve higher accuracy's when classifying activities of a new subject, the model could be retrained on a small data sample of that subject [22]. This could be done by letting the subject perform all the activities for a short amount of time and using that data to continue training the model before actually using it. Research could be conducted to test the feasibility of this method.

Currently only the window size and filter threshold were used as hyper-parameters. However, there are more parameters which could be adjusted to increase the performance of the model. Examples of these hyper-parameters are overlap in the sliding window, batch size and amount of units in the LSTM layer.

5 Responsible Research

Both collecting eye tracking data and using deep learning in real world applications are subjects which should be handled responsibly.

When collecting eye tracking data to train a model such as in this paper, one should consider that the data might reveal more than the subject intended to share. Eye tracking data was shown to be correlated with information such as biometric identity, gender, age, ethnicity, body weight, personality traits, drug consumption habits, emotional state, skills and abilities, fears, interests, and sexual preferences [23]. When collecting this data, the collector should clarify what they will use the data for and make sure it is not used for anything else. This also means that if the researcher collected the data themselves, they can not publish the data without consent of the subject. In this, the researcher should make sure what the consequences might be of publishing this data. The downside of not publishing your data as a researcher is that it is hard to reproduce your research. This means people can not validate whether the researcher's claims are correct.

Using neural networks and specifically deep learning can be dangerous when applied to some real life applications. Since AI is a black box concept, it is very hard to impossible to predict what the AI is going to do. If the AI performs an action which the maker did not expect and want it to do, dangerous situations can take place. Take for example a case where classification of gaze based activities is used to detect whether a car driver is using their phone instead of paying attention to the road. [24, 25]. If trust is put into the AI and it misclassifies the driver to be paying attention to the road, this can lead to a car accident. This means that one should never fully rely on such an AI.

References

[1] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surv.*, vol. 46, no. 3, Jan. 2014. [Online]. Available: https://doi.org/10.1145/2499621

[2] N. Srivastava, J. Newn, and E. Velloso, "Combining low and mid-level gaze features for desktop activity recognition," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 4, Dec. 2018. [Online]. Available: https://doi.org/10.1145/3287067

[3] S. Sani, N. Wiratunga, and S. Massie, "Learning deep features for knn-based human activity recognition." A. A. Sanchez-Ruiz and A. Kofod-Petersen, Eds. CEUR Workshop Proceedings, 2017, pp. 95–103. [Online]. Available: http://hdl.handle.net/10059/2489

[4] S. Sani, N. Wiratunga, S. Massie, and K. Cooper, "knn sampling for personalised human activity recognition," in *Case-Based Reasoning Research and Development*, D. W. Aha and J. Lieber, Eds. Cham: Springer International Publishing, 2017, pp. 330–344.

[5] V. Radhika, C. Prasad, and A. Chakradhar, "Smartphone-based human activities recognition system using random forest algorithm," in *2022 International Conference for Advancement in Technology (ICONAT)*, 2022, pp. 1–4.

[6] M. T. Uddin and M. A. Uddiny, "A guided random forest based feature selection approach for activity recognition," in *2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, 2015, pp. 1–6.

[7] A. Bulling, J. A. Ward, H. Gellersen, and G. Tröster, "Eye movement analysis for activity recognition using electrooculography," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 741–753, 2011.

[8] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM - a tutorial into long short-term memory recurrent neural networks," *CoRR*, vol. abs/1909.09586, 2019. [Online]. Available: http://arxiv.org/abs/1909.09586

[9] B. Rai, *Why do we use LSTM networks?* Packt Publishing, Limited, 2019.

[10] K. Kunze, Y. Utsumi, Y. Shiga, K. Kise, and A. Bulling, "I know what you are reading: recognition of document types using mobile eye tracking," Sep. 2013, pp. 113–116.

[11] G. Lan, B. Heit, T. Scargill, and M. Gorlatova, *GazeGraph: Graph-Based Few-Shot Cognitive Context Sensing from Human Visual Behavior*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 422–435. [Online]. Available: https://doi.org/10.1145/3384419.3430774

[12] G. Chevalier, "Lstms for human activity recognition," 2016.

[13] M. Ullah, H. Ullah, S. D. Khan, and F. A. Cheikh, "Stacked lstm network for human activity recognition using smartphone data," in *2019 8th European Workshop on Visual Information Processing (EUVIP)*, 2019, pp. 175–180.

[14] B. KumarSingh, K. Verma, and A. Thoke, "Investigations on impact of feature normalization techniques on classifier & performance in breast tumor classification," *International Journal of Computer Applications*, vol. 116, pp. 11–15, Apr. 2015.

[15] Z. Wu, D. Rincon, J. Luo, and P. D. Christofides, "Machine learning modeling and predictive control of nonlinear processes using noisy data," *AIChE Journal*, vol. 67, no. 4, p. e17164, 2021. [Online]. Available: https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.17164

[16] T. Boulmaiz, M. Guermoui, and H. Boutaghane, "Impact of training data size on the LSTM performances for rainfall–runoff modeling," *Modeling Earth Systems and Environment*, vol. 6, no. 4, pp. 2153–2164, Jun. 2020. [Online]. Available: https://doi.org/10.1007/s40808-020-00830-w

[17] B. Brockbernd, "Cognitive activity recognition by analyzing eye movement with convolutional neural networks," Jun. 2022, unpublished.

[18] V. Chatalbasheva, "Eye tracking-based sedentary activity recognition with conventional machine learning algorithms," Jun. 2022, unpublished.

[19] J. Meijerink, "Eye tracking-based reading activity recognition with conventional machine learning algorithms," Jun. 2022, unpublished.

[20] O. Poeth, "Eye tracking-based desktop activity recognition with conventional machine learning," Jun. 2022, unpublished.

[21] W. Koehrsen, "Overfitting vs. underfitting: A complete example," *Towards Data Science*, 2018.

[22] N. Beringer, A. Graves, F. Schiel, and J. Schmidhuber, "Classifying unprompted speech by retraining lstm nets," in *Artificial Neural Networks: Biological Inspirations – ICANN 2005*, W. Duch, J. Kacprzyk, E. Oja, and S. Zadrożny, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 575–581.

[23] J. L. Kröger, O. H.-M. Lutz, and F. Müller, *What Does Your Gaze Reveal About You? On the Privacy Implications of Eye Tracking*. Cham: Springer International Publishing, 2020, pp. 226–241. [Online]. Available: https://doi.org/10.1007/978-3-030-42504-3_15

[24] A. S. Le, T. Suzuki, and H. Aoki, "Evaluating driver cognitive distraction by eye tracking: From simulator to driving," *Transportation Research Interdisciplinary Perspectives*, vol. 4, p. 100087, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590198219300867

[25] J. Stapel, M. E. Hassnaoui, and R. Happee, "Measuring driver perception: Combining eye-tracking and automated road scene perception," *Human Factors*, vol. 64, no. 4, pp. 714–731, 2022, pMID: 32993382. [Online]. Available: https://doi.org/10.1177/0018720820959958