Research Document

Enhancing Accuracy in Machine Learning: Deep Learning Model Optimization

Company (Versuni) Mentors Signature:
Rainer Holscher

………………………

Sjaak Dijkstra

………………………

Kin Leong Yip

YIP KL
………………………

Author: Paras Khosla
Date: 15-01-24
Student no: 4289382
Department: Information & communication technology

Date:	16-01-2024
Version:	2.0.1
Authors:	Paras khosla

Version	Author	Date	Description	Status
1.0.0	Paras khosla	11-11-23	Research report created	Done
1.0.1	Paras khosla	19-12-23	Added dot framework	Done
2.0.0	Paras khosla	15-01-24	Document finalized	Done
2.0.1	Paras Khosla	16-01-24	Referenced appendix	Done

Abstract

In this research project, the primary objective was to enhance the accuracy of floor detection in a vacuum cleaner using machine learning techniques. The existing algorithm, relying on brush rotation speed, encountered difficulties distinguishing between similar floor types (figure 8). To address this, a deep learning model implemented utilizing TensorFlow and Keras, informed by insights from community engagement and expert consultations. The research covered various aspects, including data preparation techniques, strategic sequence padding and truncation, and the advantages of adopting a lower learning rate. These experiments resulted in significant accuracy improvements from 50%(figure 5) to 70%(figure 10), confirming the effectiveness of the chosen deep learning model. The research concludes with recommendations for continuous refinement, engagement with the machine learning community, and staying informed about evolving methodologies to ensure the ongoing enhancement of the developed floor detection model.

Table of Contents

Table of Figures

List of Abbreviations

DOT	Development Oriented Triangulation
STARR	Situation Task Action Result Reflection
RPM	Rotation Per Minute
ML	Machine Learning
ADAM	Adaptive Moment Estimation
RMSPROP	Root Mean Squared Propagation
VS	Visual Studio
APA	American Psychological Association

1. Introduction

In this project, the objective is to enhance the accuracy of floor detection (hard or soft floor) using machine learning techniques (Tucci & Burns, 2023). The current algorithm in the vacuum cleaner (De Philips 7000 Series Steelstofzuiger, n.d.). That was responsible for floor type detection that detects the floor via motor current based on the brush rotation on different floor types. The hard floor rpm is usually low and soft floor rpm is high based on the carpet type as there are 8 carpets (figure 8) defined by Versuni. Sometimes it misses accurate floor identification due to some carpets in the market that are quite like the hard floor, that can be seen in the figure 9. This not only affects the efficiency of cleaning but also leads to reduce the runtime of vacuum cleaner. The aim is to achieve this by creating a highly efficient machine learning model for quick floor detection based on motor current data. The project's goal is to utilize machine learning to enhance floor detection, allowing the vacuum cleaner to adjust its settings, particularly motor speed, for more effective cleaning. Focusing on utilizing motor current data, the model aims to enhance the prediction performance of floor types. In the coming weeks, primary focus will be on creating, refining, and evaluating the model. The aim to establish a machine learning solution for prompt and effective floor detection, capable of discerning various floor types (hard and soft floor).

This approach centres around the research question: How to employ advanced machine learning methods to build a smart and accurate floor detection machine learning model, surpassing the current algorithm's performance using motor current data? How can these questions help in creating a smart machine learning solution for improving the current algorithm in floor detection?

About Versuni:

Versuni is a distinguished company specializing in household appliances, headquartered in Amsterdam, the Netherlands, and serving customers in over 100 countries (*About Us*, n.d.). The wide product range includes kitchen appliances, coffee machines, air purifiers, air fryers, vacuum cleaners, and handheld steamers. With a mission to make lives healthier and happier, at Versuni, the focus is on creating tech products that not only perform smart well but also showcase a modern design. They prioritize environmental consciousness, focusing on eco-friendly and energy-efficient designs. This commitment reflects their belief in the responsibility of creating solutions that benefit consumers while positively impacting the environment. In addition to their comprehensive product lineup, Versuni specializes in advanced vacuum cleaners and handheld steamers, offering efficient and convenient solutions for the households. Their dedication to quality and innovation is evident in every aspect of their diverse range of appliances.

2. Research

In the upcoming research questions chapter, the 'DOT framework,' will be used a framework for organizing and understanding key ideas related to the research topic (Wikiwijs maken, n.d.). Acknowledging the feedback about the need for a clearer explanation of how the research framework works, this will provide a detailed overview of the various methods offered by the 'DOT framework.' This systematic approach enhances the ability to analyse concepts, contributing to a clearer understanding of research findings. By incorporating a detailed explanation of the framework's functionality, the goal is to enhance clarity and organization within the study, ensuring a complete and meaningful exploration of the subject matter and 'DOT framework' will help by organizing topic findings in a structured way that helps using a specific research method from the 'DOT framework' (Vogel, n.d.). To address the main research question, sub-research questions answers will provide answer to the main question. Furthermore, DOT framework is like a roadmap for organizing and talking about your research. It has three parts: figuring out "What" you're researching (the areas), looking into "Why" you're doing it (the choices you make), and digging into "How" you're doing it (the plans and methods).

Research Methods: "By using a bunch of methods to get the answers – it's like having different tools for different jobs. By looking at real data (Exploratory Data Analysis), checking out what's already available (Available Product Analysis), talking to experts (Expert Interview), searching online and reading a lot (Online Research and Literature Review), exploring new ideas (Exploratory Research), asking the community for their thoughts (Community Research), testing and comparing different things (A/B Testing and Comparisons), (Thorough Testing and Comparisons), looking closely at data (Exploratory Data Analysis in Machine Learning), figuring out what's good and bad (Best Good and Bad Practices), checking how things are written (Static Program Analysis), making sure things are easy to use (Usability Testing), comparing with others (Competitive Analysis), breaking down tasks (Task Analysis), and testing different parts separately (Component Test). Each method helps us get a different piece of the puzzle.

2.1 Research questions

Main Research Question:

> Q: What are the most effective approaches for optimizing the performance of a machine learning model?

Sub-Research Questions:

1. Q: What are the reasons for opting for a deep learning model?

Research Insights: The choice to use a deep learning model was carefully based on the research, considering various aspects and factors:

I. **Community Research:**

- Engaging with the community, I actively participated in forums, discussions, and online communities dedicated to machine learning. This facilitated the gathering of insights and recommendations from practitioners who encountered similar challenges (Grossfeld, 2023).

- Platforms such as Stack Overflow and machine learning communities on Reddit proved invaluable for accessing collective knowledge and real-world experiences.

II. **Expert Interview:**

- Engaging in discussions with domain expert, these conversations offered valuable insights into the suitability of deep learning models for specific project requirements like based on the time series signals data (Sivanesan, 2023, Personal interview).

- Interactions with expert helped to take decisions on choosing a specific machine learning model and classical machine learning model.

III. **A/B Testing and Comparisons:**

- Adapting the A/B testing methodology, I conducted systematic comparisons of different methods to ascertain the most effective approach. This hands-on exploration allowed for an empirical understanding of the performance implications of various model choices like deep learning vs classical machine learning.

- Platforms such as GitHub served as valuable resources for accessing diverse datasets and benchmarking models (Martiniblack, n.d.).

- It also helped me to find out which tool is best while working with these models.

2. Q: How was the decision to use TensorFlow and Keras for the deep learning model supported?

Research Insights: The selection of TensorFlow and Keras for the deep learning model was underpinned by a comprehensive research methodology, combining an extensive literature review, expert consultations, and an available product analysis. The multi-faceted approach aimed to ensure a well-informed decision in line with project requirements and industry best practices.

I. **Literature Review:**

- A literature review was conducted to gather insights from scholarly articles and publications related to deep learning frameworks (Sharma, 2022).

- Existing knowledge was explored to understand prevailing practices and advancements in the field, providing a foundation for informed decision-making.

- Official documentation links for TensorFlow and Keras explored to delve into technical details and features (Machine learning for beginners and experts, n.d.; Team, K., n.d.).

II. **Expert Interview:**

- Discussions with domain expert played an important role in the decision-making process. Insights from experienced practitioner provided valuable perspectives on the suitability of TensorFlow and Keras for specific deep learning project requirements (Sivanesan, 2023, Personal interview).

- Expert opinions contributed practical considerations, ensuring alignment with learning prospective and best practices.

III. **Available Product Analysis:**

- An analysis of available products involved assessing the features, functionalities, and suitability of TensorFlow and Keras for the deep learning model.
- This analysis considered the official documentation, community support, and user experiences to evaluate the practical aspects of each tool (Reddiculess, n.d.).

3. Q: Why opt for the 'verbose,' 'epochs,' and 'batch_size' functions in my deep learning ML model?

Research Insights: To ascertain the most effective parameters for optimizing my deep learning model, I employed a diverse range of research methods. The methodology involved:

I. **Literature Review:**

- An extensive literature review was conducted to explore common practices and recommendations in the field of machine learning, shedding light on established guidelines for parameter tuning (Devansh, 2023).
- Resources such as research papers, articles, and official documentation provided theoretical insights into the impact of parameters on model performance (Brownlee, J., 2022).
- Links to seminal works in machine learning literature, like Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville, explored for foundational knowledge (Goodfellow, Bengio, & Courville, n.d.).

II. **Expert Interview:**

- Insights from expert consultations helped in gaining valuable perspectives on the significance of key parameters, including 'verbose,' 'epochs,' and 'batch_size' (Pramoditha, 2023).
- Conversations with experienced practitioners contributed practical considerations, emphasizing the real-world implications of parameter choices (Sivanesan, 2023, Personal interview).
- Expert recommendations from forums like Stack Overflow and Towards Data Science considered for detailed understandings (Towards Data Science, 2001).

III. Thorough Testing and Comparisons:

- Rigorous testing and comparisons conducted to assess the impact of different parameter configurations on the model's accuracy and training efficiency.
- Multiple experiments designed, altering parameters systematically to observe their effects on the model's performance.
- Open-source frameworks like TensorFlow utilized for visualizing training metrics and comparing different runs (TensorFlow. n.d.)

4. Q: How to set up Keras and TensorFlow on a Windows system?

Research Insights: To facilitate the installation of Keras and TensorFlow on Windows, by employing a diverse range of research methods. The methodology involved:

I. **Community Research:**

- Delving into community research, the forums, discussion boards, and online communities visited to gather insights and recommendations from users who encountered similar installation challenges on Windows. (FSantosCodes, n.d.).
- Valuable discussions on platforms like Stack Overflow and Reddit provided practical tips and solutions shared by the community.

II. **Exploratory Data Analysis (Machine Learning):**

- Conducting an exploratory data analysis (Machine Learning), I systematically examined common installation issues and solutions encountered by the machine learning community on Windows, such as facing a 'pip not found' error (Stack Overflow, n.d.)
- Analysing user-reported problems and resolutions in forums and GitHub repositories, such as the TensorFlow GitHub Issues (RStudio. n.d.), revealed patterns and potential workarounds.

III. **Best Good and Bad Practices:**

- Utilizing best good and bad practices, I consulted expert opinions and implemented procedures to ensure a reliable and effective installation process.
- Recommendations from authoritative sources, including the official documentation of TensorFlow and Keras, incorporated to follow established best practices. To install TensorFlow 2, the official documentation provides detailed instructions (TensorFlow, n.d.). Additionally, the Keras Documentation by the Keras team offers valuable insights into Keras (Keras Team, n.d.).

5. Q: How does the padding and truncating of sequences contribute to data preparation?

Research Insights: These three methods collectively provide a comprehensive understanding of how padding and truncating sequences contribute to data preparation:

I. **Literature Study:**

- A comprehensive review of relevant literature on sequence data preprocessing in machine learning underscores the significance of padding and truncating sequences. Research articles, such as "Preparing text data for transformers: tokenization, mapping and padding" by (Lokare, G., 2023)., emphasize the necessity for input sequences to maintain consistent lengths to facilitate effective model training.
- Padding ensures that sequences of varying lengths can be processed by the model by adding zeros or other tokens to achieve a uniform length. Truncating, on the other hand, helps handle excessively long sequences that might hinder computational efficiency during training.

II. **Exploratory Data Analysis (Machine Learning):**

- In the context of exploring data for deep learning model development, it is essential to investigate the distribution of sequence lengths within the dataset. This exploration provides valuable insights into whether padding and truncating are necessary. When working with sequences, such as in deep learning model where you're applying padding and truncating, libraries like Matplotlib can be effectively used to visualize the distribution of sequence lengths. Understanding the distribution can help to make informed decisions about the appropriate length for padding and truncation, ensuring that the model processes sequences effectively (Figure1).
- Analysing the dataset may reveal variations in sequence lengths, and the decision to pad or truncate can be based on maintaining a balance between retaining crucial information and optimizing computational efficiency.

III. **Community research:**

- Articles by machine learning practitioners and experts, such as Sujatha Mudadla, a seasoned data scientist, provided insights into practical considerations regarding sequence data preprocessing. In her article, Mudadla highlighted the significance of padding to ensure consistent input sizes during training (Mudadla, S., 2023).
- Brownlee highlighted the importance of truncation in situations where excessively long sequences could create memory challenges. Insights from the article provide valuable real-world perspectives, shedding light on the considerations and decision-making processes associated with padding and truncating sequences (Brownlee, 2019).

6. Q: How does the code ensure a fair comparison by repeating the experiment multiple times?

Research Insights: These three methods collectively provide a comprehensive understanding of how repeating the experiment multiple times contributes to fair and reliable comparisons:

I. **A/B Testing:**

- Conducting A/B testing in the experimental setup is crucial for validating the fairness of comparisons. By systematically varying conditions or parameters across multiple iterations, A/B testing helps mitigate the impact of confounding variables and random biases.
- Randomizing the order in which experiments are conducted and ensuring a balanced distribution of test cases across variations contribute to the statistical robustness of the comparison. This method allows for a examination of the consistency of results across different runs.

II. **Literature Study:**

- An examination of relevant literature on experimental design and statistical analysis reinforces the necessity of repeated experiments for achieving credible outcomes. Articles like (Stack exchange, n.d.), emphasize the importance of replication for building confidence in research findings.
- Literature also highlights the impact of external factors, and repetition serves as a means to account for variability and validate the generalizability of results. It

establishes a foundation based on established scientific principles, contributing to the credibility of the comparison methodology.

III. **Expert Interview:**

- Conducting in-depth interviews with Sudhakar, a seasoned expert in experimental design and statistical analysis, provides a qualitative understanding of the significance attached to repeated experiments. Sudhakar can offer nuanced insights on the practical challenges and considerations associated with ensuring fairness in comparisons (Sivanesan, 2023, Personal interview).

- Expert interview allows for a more personalized exploration of factors influencing the decision to repeat experiments, providing context-specific recommendations. These insights enrich the research methodology by incorporating practical wisdom from Sudhakar's experienced professional perspective.

7. Q: How are the overall predictions printed, and what insights can be derived from the predictions?

Research Insights: the research exploration delves into three key methods. First, static program analysis unveils the underlying code structure governing the printing of predictions. Following this, machine learning-focused exploratory data analysis sheds light on the model's decision-making and presentation of probability scores. Lastly, usability testing emerges as a crucial element, involving real-world user feedback to assess the user-friendliness and effectiveness of the displayed predictions:

I. **Static Program Analysis:**
- Examining the code structure reveals a systematic approach to printing overall predictions. The use of static program analysis, specifically reviewing the code implementation, elucidates the step-by-step process involved in displaying predictions. The code employs arrays to present probabilities for each class, indicating a standardized format consistent with established programming practices.

II. **Exploratory Data Analysis (Machine Learning):**

- Leveraging exploratory data analysis techniques specific to machine learning, the printed predictions unveil insights into the model's decision-making. The probability scores for "Hard Floor" and "Soft Floor" classes, displayed sequentially, offer a glimpse into the model's confidence levels. Visualizing these scores through tools like Matplotlib could further enhance the exploration, allowing for a comprehensive understanding of the distribution and patterns in the prediction data (How does the model make predictions? | Allgaier, J., 2023).

III. **Usability Testing:**

- Usability testing involves gathering feedback from users to assess the effectiveness and user-friendliness of a system or product. Applying usability testing to the printed predictions allows for real-world insights into how users interpret and find value in the displayed information. Test participants can provide feedback on the clarity, comprehensibility, and overall usability of the prediction format, ensuring that the information meets user expectations and facilitates meaningful decision-making.

8. Q: What is the purpose of using the Adam optimizer with a specified learning rate?

Research Insights: These three methods used for the research about using Adam optimizer in the model:

I. **Expert Interview:**
   - Insights from discussions with optimization algorithm experts, emphasize the strategic role of the Adam optimizer with a specified learning rate.
   - Machine learning expert highlights Adam's adaptive learning rate capability, contributing to efficient convergence in complex optimization landscapes (Sivanesan, 2023, Personal interview).
   - The specified learning rate is identified as a crucial tuning parameter, offering a balance between rapid convergence and stability during training.
   - Practical considerations and nuanced insights from the expert interview with Sudhakar provide a deeper understanding of the Adam optimizer's role in achieving optimal model performance.

II. **Literature Study:**
   - A comprehensive literature review on optimization algorithms and learning rate strategies reaffirms the importance of the Adam optimizer with a specified learning rate.
   - Notable studies, such as "Optimization Algorithms in Deep Learning: A Comprehensive Review" by (Brownlee, J., 2021). highlight Adam's adaptive nature in adjusting learning rates based on historical gradients.
   - The specified learning rate emerges as a critical factor influencing the algorithm's ability to navigate diverse and complex optimization landscapes.
   - The literature study establishes a theoretical foundation, consolidating insights from the sources to validate the chosen optimization approach.

III. **Competitive Analysis:**
   - Observing the prevalence of the Adam optimizer with specified learning rates in state-of-the-art models across competitive benchmarks enhances the understanding of its effectiveness.
   - Identifying patterns and preferences in optimization choices within the competitive landscape validates the strategic importance of the specified learning rate in optimizing model convergence and performance.
   - Competitive analysis complements theoretical and expert-driven insights, offering a real-world perspective on the widespread adoption of the Adam optimizer with a specified learning rate in contemporary deep learning models.

9. Q: How softmax is helping the deep learning model?

Research Insights: Utilizing a combination of insights from expert, literature studies, and exploratory data analysis in machine learning, by delve into understanding how **softmax** contributes to the effectiveness of deep learning model:

I. **Expert Interview:**

- In discussions with leading expert in deep learning, it becomes evident that softmax plays a pivotal role in enhancing the interpretability of the deep learning model's outputs.
- Sudhakar emphasizes that softmax transforms the model's raw output scores into probability distributions, providing clear and normalized class probabilities.
- The expert interview underscores how the use of softmax aids in making informed decisions based on the model's predictions, especially in multi-class classification scenarios (Sivanesan, 2023, Personal interview).

II. **Literature Study:**

- A comprehensive literature study on softmax activation functions reaffirms its significance in deep learning models.
- Studies, such as "**Softmax Activation Function with Python**" by (Brownlee, J.,2020), highlight all the details about the softmax use.
- The literature study establishes that softmax not only facilitates effective decision-making but also contributes to model generalization and stability during training.

III. **Exploratory Data Analysis (ML):**

- Employing exploratory data analysis techniques specific to machine learning involves examining the impact of softmax on the model's predictions.
- Visualization of the softmax output probabilities across various instances in the dataset provides insights into how the model assigns confidence scores to different classes.
- This empirical exploration reveals the smoothing effect of softmax (AiOTA LABS, 2021), ensuring that the model's predictions are well-calibrated and aiding in understanding the model's uncertainty in different prediction scenarios.

10. Q: why we are using hot encoding in deep learning model and how it is helping?

Research Insights: The research methods, combining insights from expert, established theories, and specific task considerations, underscores the strategic importance of implementing one-hot encoding in the deep learning model:

I. **Expert Interview:**

- Seeking guidance from experienced data scientist, provides valuable insights into the strategic application of one-hot encoding in the deep learning model.
- He highlights the fundamental role of one-hot encoding as a preprocessing step, ensuring the compatibility of categorical data with the model's input specifications and improving its capacity to differentiate between distinct classes (Sivanesan, 2023,

Personal interview).

- Expert opinions emphasize that one-hot encoding is an essential practice for successful model training and prediction, especially in situations involving diverse categorical classes.

II. **Literature Study:**

- An exploration of relevant literature on deep learning and categorical data processing supports the ubiquitous use of one-hot encoding.
- Studies, like "why one-hot encoding " by (Dey, V., 2021), underscore that one-hot encoding provides a standardized approach for incorporating categorical variables into neural network architectures.
- The literature study reinforces that one-hot encoding promotes model flexibility and interpretability in handling diverse categorical information.

III. **Task Analysis:**

- Conducting a task analysis specific to the objectives of the deep learning model involves understanding the role of categorical variables in the overall task.
- The task analysis reveals that one-hot encoding aligns with the model's objective, ensuring that categorical distinctions are accurately represented and leveraged during training and inference.
- Task-specific considerations emphasize the importance of one-hot encoding for preserving meaningful category relationships within the context of the intended use case.

11. Q: How is it helping to use a lower learning rate for the model?

Research Insights: This research approach, combining insights from literature, expert opinions, and empirical analysis, aims to provide a comprehensive understanding of the advantages associated with using lower learning rates in model training:

I. **Literature Study:**

- Review existing literature on machine learning optimization algorithms and learning rates.
- Explore studies such as "How it improves performance in deep learning" (Zulkifli, H., 2019, March 28) for theoretical foundations and empirical evidence supporting the advantages of lower learning rates.

II. **Expert Interview:**

- Conducted interview with machine learning practitioners and optimization experts, like Sudhakar.
- Gather practical perspectives, real-world scenarios, and best practices concerning the selection of lower learning rates (Sivanesan, 2023, Personal interview).

III. **Component Test:**

- Conduct focused experiments to evaluate the impact of different learning rates on individual model components.
- Isolate factors such as convergence speed, stability, and generalization for systematic observation.
- Explore how lower learning rates influence the performance of specific components.
- Identify the optimal learning rate through granular testing for improved model training efficacy.

12. Q: How to achieve higher accuracy for the machine learning model?

Research Insights: This multi research approach combines literature insights, A/B testing, and data analytics help to address the goal of achieving higher accuracy for the machine learning model:

I. **Literature Study:**

- Conduct an extensive literature study to review existing research papers, articles, and publications focused on methods to enhance machine learning model accuracy.
- This involves synthesizing knowledge from the academic community to understand the latest advancements and proven strategies.

II. **A/B Testing:**

- Implement A/B testing to systematically compare different configurations, hyperparameters, or preprocessing techniques for the machine learning model.
- By experimenting with variations and measuring their impact on accuracy, A/B testing allows for data-driven decision-making to optimize model performance.

III. **Data Analytics:**

- Utilize data analytics to analyse the dataset, identify patterns, and discover correlations that can inform strategies for improving accuracy.
- Data-driven insights obtained through analytics contribute to informed decision-making in model refinement.

13. Q: Why is 50% accuracy not helping the model and not satisfying the requirements?

Research Insights: By combining insights from model evaluation, exploratory data analysis, and task analysis, a detailed understanding of why a 50% accuracy rate is not contributing to model improvement will be obtained:

I. **Model Evaluation (Machine Learning):**

- Conducting a detailed evaluation of the model's performance metrics, including accuracy, precision, recall, and prediction score.
- Analysing these metrics provides insights into the specific areas where the model may be struggling, allowing for targeted improvements.

II. **Exploratory Data Analysis** (**Machine Learning**):

- Undertaking exploratory data analysis to identify patterns and anomalies in the dataset that might be hindering model performance.

- This involves a detailed examination of the distribution of features, handling outliers, and understanding the characteristics of the data.

III. **Task Analysis:**

- Conducting a task analysis to evaluate whether the chosen machine learning task aligns with the model's capabilities.

- This involves scrutinizing the nature of the problem, the complexity of the data, and whether the model is suitable for the given task.

14. Q: Which technologies and tools are most suitable for implementing and deploying machine learning models for floor detection?

Research Insights: This complete approach, which includes studying relevant literature, checking community discussions, and conducting practical tests, helps us fully understand the best technologies and tools for floor detection applications:

I. **Literature Study:**

- Conduct a detailed study of existing literature on machine learning models for floor detection.

- Review research papers, articles, and case studies to identify technologies and tools commonly utilized. This method helps in understanding the theoretical foundations and established practices in the field.

II. **Community Research:**

- Explore online forums, community discussions, and social platforms where practitioners share their experiences.

- Analyzing user discussions to identify popular technologies, tools, and emerging trends in the implementation and deployment of machine learning models for floor detection.

III. **Benchmark Test:**

- Perform benchmark tests to evaluate the performance of different technologies and tools in the context of floor detection.

- Compare their efficiency, accuracy, and scalability to identify the most suitable options for implementing and deploying machine learning models.

15. Q: How subtracting with the first 40 samples of median in the dataset helped in increasing efficiency?

Research Insights: By using methods like data exploration, analytics, expert interviews, and benchmark tests, we can gain a full understanding of how subtracting the median of the initial 40 samples improves efficiency:

I. **Data Analytics:**

- Utilize data analytics techniques to quantitatively assess the efficiency gains achieved by subtracting the median of the first 40 samples.

- This involves performing statistical tests, calculating relevant metrics, and comparing the performance of models or algorithms using the modified dataset against the original dataset.

II. **Expert Interview:**

- Conduct expert interviews with data scientists or domain experts who have experience in data preprocessing and feature engineering.
- Experts can provide qualitative insights into the rationale behind subtracting the median of the first 40 samples, potential advantages, and scenarios where this technique is most beneficial.

III. **Benchmark Test:**

- Implement benchmark tests to evaluate the efficiency of models or algorithms using the modified dataset against a baseline or standard dataset without the subtraction.
- This comparative analysis helps quantify the improvement in efficiency and assess the practical implications of the subtraction process.

3. Results

1. Q: What are the reasons for opting for a deep learning model?

Ans: The decision to employ a deep learning model was grounded in a comprehensive research approach that considered various aspects and factors. Community research, engagement with experts, A/B testing, and systematic comparisons were key components of this decision-making process. Active participation in machine learning forums, such as Stack Overflow and Reddit, facilitated the gathering of insights and recommendations from practitioners who faced similar challenges (Martiniblack, n.d.). Expert consultations provided valuable insights into the suitability of deep learning models for the specific project requirements, such as working with time series signals data.

2. Q: How was the decision to use TensorFlow and Keras for the deep learning model supported?

Ans: The selection of TensorFlow and Keras for the deep learning model was underpinned by a comprehensive research methodology, combining an extensive literature review, expert consultations, and an available product analysis. The literature review involved a examination of scholarly articles related to deep learning frameworks (Sharma, 2022). Expert consultations played a crucial role in gaining valuable perspectives on the suitability of TensorFlow and Keras for specific project requirements. The available product analysis assessed the features, functionalities, and suitability of both tools for the deep learning model, considering factors such as official documentation, community support, and user experiences (Reddiculess, n.d.).

3. Q: Why opt for the 'verbose,' 'epochs,' and 'batch_size' functions in my deep learning ML model?

Ans: While working on deep learning models, the 'verbose,' 'epochs,' and 'batch_size' functions are crucial parameters that influence the training and performance of the model:

- **Verbose:** The 'verbose' parameter determines the amount of information printed during the training process. It controls whether to display progress bars, logging information, or no information at all. Choosing an appropriate level of verbosity depends on the user's preference and the need for real-time updates on the training progress.

- **Epochs:** The 'epochs' parameter signifies the number of times the entire training dataset is passed forward and backward through the neural network. One epoch equals one complete presentation of the dataset to the model. The choice of the number of epochs is essential as it affects how well the model learns from the data. Too few epochs may result in underfitting, while too many epochs may lead to overfitting (Epochs, Batch Size, Iterations - How they are Important, n.d.).

- **Batch Size:** The 'batch_size' parameter determines the number of samples processed before updating the model's weights. Training neural networks on the entire dataset (appendix C) at once can be computationally expensive. Batch training, where the dataset is divided into smaller batches, allows for more frequent weight updates. The optimal batch size depends on factors like available memory, computational resources, and the nature of the dataset (Devansh, 2023).

These three parameters collectively influence the efficiency, performance, and resource utilization of a deep learning model during the training phase. Adjusting them appropriately is essential for achieving the desired balance between models' accuracy. The code snippet can be found at Figure 3.

4. Q: How to set up Keras and TensorFlow on a Windows system?

Ans: To set up Keras and TensorFlow on the Windows system for use in Jupyter Notebook:

1. Install Anaconda: Download and install Anaconda, which includes Python (*Installation — Anaconda documentation*, n.d.).
2. Open Anaconda Navigator: Launch Anaconda Navigator after installation.
3. Create a New Environment: Set up a new environment within Anaconda.
4. Install Jupyter Notebook: Install Jupyter Notebook within the new environment.
5. Open Jupyter Notebook: Launch Jupyter Notebook.
6. Install Keras and TensorFlow:
    - Create a new notebook in Jupyter.
    - Use a cell to run the following commands:

    Follow the step 1 in Figure 4

    - Verify Installation

    In a new cell, run:
    Follow the step 2 in Figure 4

These steps will ensure that Keras and TensorFlow are successfully installed and ready to use in the Jupyter environment.

5. Q: How does the padding and truncating of sequences contribute to data preparation?

Ans: Padding and truncating sequences are essential practices in data preparation for machine learning models. A literature study underscores the significance of these methods for effective sequence data preprocessing. Padding involves adding zeros or tokens to maintain consistent sequence lengths, ensuring uniformity for model training (Lokare, G., 2023). In exploratory data analysis (ML), investigating the distribution of sequence lengths using tools like Matplotlib is vital. Expert reviews, such as insights from Sujatha Mudadla, emphasize the practical considerations of padding to ensure consistent input sizes during training (Mudadla S., 2023). Balancing the retention of crucial information with computational efficiency guides decisions on whether to pad or truncate sequences.

6. Q: How does the code ensure a fair comparison by repeating the experiment multiple times?

Ans: In the results, the code ensures a robust comparison by meticulously repeating the experiment multiple times. This deliberate repetition is a thoughtful strategy to boost the confidence and dependability of the results (Stack exchange, n.d.). By carrying out the experiment numerous times, the code minimizes the impact of any chance variations or errors that might occur in a single run. The averaging of results across these repetitions is a systematic way to offer a complete and clearer insight into the effectiveness of the methods used. This rigorous approach is designed to bolster the reliability and consistency of the findings, thereby strengthening the trustworthiness of the conclusions derived from the experiment.

7. Q: How are the overall predictions printed, and what insights can be derived from the predictions?

Ans: The overall predictions are printed for each label, providing insights into the model's classification for "Hard Floor" and "Soft Floor." The predictions display the likelihood scores associated with each class. For instance, in the "Label_0: Hard Floor" category, the model assigns scores like [0.5459894, 0.5459894, ... 0.5459894], and for "Label_1: Soft Floor," it assigns scores like [0.45401064, 0.45401064, ... 0.45401064]. The values in these arrays represent the probability of the corresponding class. Higher values indicate a higher likelihood of belonging to that class. The insights derived from these predictions include understanding the model's confidence in its classifications and identifying potential areas for improvement in cases where the scores are close. The presented accuracy, such as "Accuracy: 50.554% (+/-0.078)," reflects the overall performance of the model across multiple runs, considering the variability in predictions (Allgaier, J., Mulansky, L., Draelos, R. L., & Pryss, R., 2023). It indicates the model's ability to make correct classifications on the test dataset(Appendix c) The results can be seen in the (Figure 5).

8. Q: What is the purpose of using the Adam optimizer with a specified learning rate?

Ans: In the results section, the impact of employing the Adam optimizer with a specified learning rate is evident in the training dynamics of the deep learning model. The code snippet below highlights the configuration of the optimizer with a specific learning rate can be seen in (Figure 6). The **Adam** optimizer is instantiated with the specified learning rate (**learning_rate=0.001**). This setting allows the model to dynamically adjust its learning rate during training, contributing to effective convergence. The impact of this configuration becomes apparent in the training history (**history**), showcasing the model's performance over epochs. The specified learning rate, combined with the Adam optimizer, plays a crucial role in achieving optimal training outcomes (Brownlee, J., 2021).

9. Q: How **softmax** is helping the deep learning model?

Ans: In deep learning model, the incorporation of the **softmax** activation function emerges as a crucial element, contributing significantly to model performance. Through expert interview, it becomes evident that **softmax** enhances interpretability by transforming raw output scores into probability distributions. A literature study affirms the role of **softmax** in ensuring normalized class probabilities, aiding in effective decision-making. Empirical results from exploratory data analysis showcase the practical impact of **softmax**, revealing its smoothing effect on prediction probabilities and contributing to a well-calibrated model. Overall, **softmax** proves instrumental in improving model interpretability, decision-making, and generalization (AiOTA LABS., 2021).

10. Q: why we are using hot encoding in deep learning model and how it is helping?

Ans: In the context of the deep learning model, the incorporation of one-hot encoding holds paramount importance for the effective management of categorical data. One-hot encoding serves as a critical preprocessing technique, ensuring that categorical variables are accurately represented in a format compatible with the model's input requirements. This strategic decision is substantiated by insights drawn from expert interviews, theoretical underpinnings gleaned from literature studies, and specific considerations tailored to the tasks at hand. Through extensive testing and empirical analysis, the results Confirm the important role of one-hot encoding in enhancing the model's discriminative capabilities among various categorical classes (Dey, V., 2021). This, in turn, contributes to a notable improvement in overall accuracy and the model's ability to make robust predictions. The thoughtful integration of one-hot encoding emerges as a key factor in optimizing the model's performance and proficiently handling the complexities.

11. Q: How is it helping to use a lower learning rate for the model?

Ans: Utilizing a lower learning rate for the model proves advantageous as it contributes to a more stable and precise training process. It also helps where the data is time series signals so for this case there are high and low signals from the motor, it is useful for us using a lower learning rate. A lower learning rate allows the model to incrementally adjust its parameters, enabling finer convergence towards the optimal solution. This measured approach helps prevent overshooting and oscillations during training, particularly in scenarios with complex or noisy datasets. The use of a lower learning rate is often instrumental in achieving better generalization, ensuring that the model's learned patterns extend effectively to unseen data. This choice promotes a smoother optimization trajectory, enhancing the model's overall robustness and performance in capturing intricate patterns within the data (Zulkifli, H., 2019). The coding snippet can be seen in the Figure 7.

12. Q: How to achieve higher accuracy for the machine learning model?

Ans: Achieving higher accuracy for the machine learning model involves an approach that combines literature findings, A/B testing, and data analytics. By researching and implementing proven methodologies, fine-tuning hyperparameters, and exploring feature selection techniques, can enhance the model's performance. Engaging in literature study ensures alignment with the latest advancements in the field. A/B testing enables systematic comparison of different configurations like improving the learning rate that helps the model to learn in more detail as we are using time series signals for the model that is helping to increase the efficiency, and data analytics helps uncover patterns within the dataset (appendix C) that done during data collection using different scenarios (low, medium high heights), guiding informed decisions for optimizing accuracy. This holistic research strategy aims to elevate the model's overall effectiveness in making accurate predictions.

13. Q: Why is 50% accuracy not helping the model and not satisfying the requirements?

Ans: Achieving a 50% accuracy (figure 5) in model predictions is deemed unsatisfactory and falls short of meeting the requirements due to its equivalence to random chance. In machine learning, a model with 50% accuracy essentially makes predictions as accurate as a coin flip, offering no judgement between classes. With 50% accuracy we can retrieve nothing as it can be soft or hard floor as our required target is to touch 95% accuracy or at least around 70%. Such performance renders the model ineffective for its intended purpose, as it fails to provide meaningful results or contribute to reliable decision-making. The model should significantly surpass this baseline accuracy, demonstrating a capacity to learn and generalize patterns within the data. It's target to aim for higher accuracy to ensure the model works effectively and reliably in real-world situations.

14. Q: Which technologies and tools are most suitable for implementing and deploying machine learning models for floor detection?

Ans: By using the research methods like literature search, benchmark test and community research helped to make decision regarding choice for using technologies in the project that will be for working on machine learning models for floor detection, Jupyter Notebook, TensorFlow, and Keras emerge as highly suitable technologies and tools. Jupyter Notebook (Klingler, N., 2023)., with its interactive and collaborative environment (Figure 2)., facilitates seamless code development, experimentation, and visualization, making it an ideal choice for model development. TensorFlow and Keras (Terra, J., 2023), renowned for their robustness and ease of use, serve as powerful frameworks for building and deploying machine learning models (Mahto, P., 2021). Their comprehensive libraries and efficient neural network implementations provide the necessary tools for creating accurate floor detection models. This combination of Jupyter Notebook, TensorFlow, and Keras offers an effective solution that meets the various needs of developing and deploying machine learning models for floor detection.

15. Q: How subtracting with first 40 samples of median in the dataset helped in increasing efficiency?

Ans: Subtracting the median (figure 12) of the first 40 samples in the dataset (Appendix C) has proven to enhance efficiency. This preprocessing step contributes to a more focused dataset, reducing potential outliers and aligning the data with the central tendency. By cantering the dataset around its median and excluding initial variations, the model's performance improves, leading to enhanced efficiency in tasks such as modelling and prediction. This subtraction process helps in creating a more robust and streamlined dataset, ultimately contributing to the overall effectiveness of the analysis. The code snippet that helped to subtract the 40 samples from the dataset can be found at figure 11.

4. Conclusion and recommendation’s

To conclude, after a complete research process involving exploring community articles, consulting with experts, conducting A/B tests, and making systematic comparisons, a deep learning model was chosen for implementation. Actively participating in machine learning forums helped us gather useful insights, and expert consultations guided us in handling time series signals data. The choice of TensorFlow and Keras for the deep learning model was based on a detailed review of literature, expert advice, and an analysis of available tools. We carefully selected the parameters 'verbose,' 'epochs,' and 'batch_size,' impacting the model's efficiency during training. Padding and truncating sequences were crucial for proper data preparation, ensuring consistent sequence length and efficient computation. The code ensured fair comparisons through repeated experiments, and predictions gave us insights into the model's confidence and areas to improve. The working of whole system can be seen at figure 13. Using a lower learning rate proved beneficial for stable and precise training, preventing errors, and improving generalization. Moreover, making specific adjustments significantly improved the model's performance. Switching the optimizer function from Adam to ‘Rmsprop’ played a key role, boosting accuracy from 50% to 70%. Additionally, removing the first 40 samples from the dataset collected via vacuum cleaner further enhanced the model's efficiency. These tweaks were crucial in achieving better results. Incorporating one-hot encoding significantly improved the model's ability to distinguish between categorical classes. To enhance the model further, ongoing community engagement, periodic expert consultations, and exploring new methodologies could contribute to refining its performance and adaptability.

In terms of recommendations, refining the deep learning model could involve further tuning of parameters like 'verbose,' 'epochs,' and 'batch_size' to optimize training efficiency. Continued research and expert consultations remain essential for staying alongside of evolving best practices and incorporating the latest advancements in the model tuning. Additionally, exploring alternative deep learning frameworks beyond TensorFlow and Keras might offer insights into potential enhancements or alternative approaches. Further experimentation with different learning rates could contribute to identifying the optimal balance in the training process. Continual monitoring of sequence data distribution and adapting padding/truncating strategies based on evolving datasets (appendix C) can enhance the model's adaptability. Regular updates to the model architecture and continuous validation against diverse datasets could improve its generalization capabilities. The dataset quality also matters like having proper timestamps during the data is collected it helps to increase the quality of the model training. Always have proper names of the specific datasets that helps to increase the human efficiency while working on machine learning and decrease the chances of error in the dataset. Always print the important steps of the results that helps to understand the next results prediction. Moreover, changing ‘batch_size’ value also helps to increase model quality, there’s not a specific value but we can only test by increasing or decreasing it. We can also add more filters in the model, but it depends when, while I was working it didn’t help that much. Lastly, staying involved in forums and articles related to machine learning to gain experiences and insights from the broader machine learning community could contribute to ongoing improvements.

5. Personal Reflection

I'll be going to use the STARR framework (Janse, 2023) for my personal reflection, a helpful framework that guides me in exploring different aspects of my experience. STARR stands for Situation, Task, Action, Result, and Reflection. It's like a roadmap that helps me break down and analyse specific moments or challenges I've faced. First, I describe the situation or context, then talk about the task at hand, followed by the actions I took. After that, I delve into the results of my actions and wrap it up with thoughtful reflections on what I've learned. It's a straightforward and organized way to reflect on my experiences and gain insights for personal and professional growth.

Embarking on the journey of developing and optimizing a deep learning model has been a fulfilling yet challenging experience. Throughout this process, I've gained a deeper understanding of the intricate factors influencing the model's performance. Engaging with the machine learning community on platforms like Stack Overflow and Reddit provided valuable insights, and expert consultations played a crucial role in shaping my understanding of deep learning models for specific project requirements. A/B testing and systematic comparisons allowed me to fine-tune parameters iteratively. The decision to use TensorFlow and Keras was rooted in a complete research methodology, involving literature reviews, expert consultations, and available product analyses. Experimenting with parameters such as 'verbose,' 'epochs,' and 'batch_size' unveiled their nuanced impact on training dynamics. Exploring the significance of padding and truncating sequences highlighted the delicate balance between information integrity and computational efficiency. Additionally, delving into the Adam optimizer, softmax activation function, and one-hot encoding broadened my understanding of model behaviour. Utilizing a lower learning rate for the model and changing the optimizer function from Adam to Rmsprop provided valuable insights into the importance of precision and stability in training. Reflecting on this project, it not only enhanced my technical skills but also deepened my appreciation for the collaborative and evolving nature of the machine learning field. Building, testing, and refining the model underscored the iterative and dynamic essence of the data science journey. Moving forward, I am excited about applying these insights to future projects and contributing to the ever-evolving landscape of machine learning.

6. References

DOT-Framework (EN) - Wikiwijs maken. (n.d.). wikiwijs.https ://maken.wikiwijs.nl/129804/DOT_framework__EN_#!page-4594577

Brownlee, J. (2022, august 15). *Difference between a batch and an epoch in a neural network*. MachineLearningMastery.com. https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/

Brownlee, J. (2019, 14 augustus). *Data preparation for variable length input sequences*. MachineLearningMastery.com. https://machinelearningmastery.com/data-preparation-variable-length-input-sequences-sequence-prediction/

Existing-Kale. (n.d.). *IDLE, VS Code or Jupyter notebook for learning python for data analysis? : r/learnpython*. https://www.reddit.com/r/learnpython/comments/zpkb1o/idle_vs_code_or_jupyter_notebook_for_learning/

Grossfeld, B. (2023, July 10). Deep learning en zelflerende systemen: wat is het verschil? *Zendesk NL*. https://www.zendesk.nl/blog/machine-learning-and-deep-learning/#georedirect

Goodfellow, Bengio, & Courville, n.d. *Deep learning*. (n.d.). https://www.deeplearningbook.org/

Devansh. (2023, 5 may). How does batch size impact your model Learning - geek culture - medium. *Medium*. https://medium.com/geekculture/how-does-batch-size-impact-your-model-learning-2dd34d9fb1fa

I'm trying to dowloand install Tensorflow in Python 3.8. (n.d.). Stack Overflow. https://stackoverflow.com/questions/77693569/im-trying-to-dowloand-install-tensorflow-in-python-3-8

Lokare, G. (2023, 12 February). Preparing text data for transformers: tokenization, mapping and padding. *Medium*. https://medium.com/@lokaregns/preparing-text-data-for-transformers-tokenization-mapping-and-padding-9fbfbce28028#:~:text=3.-,Padding%20%2F%20Truncation,all%20have%20the%20same%20length.

Martiniblack. (n.d.). *GitHub - martiniblack/UCI-HAR: ML Project for Human Activity Classification*. GitHub. https://github.com/martiniblack/UCI-HAR

Mudadla, S. (2023, 20 October). What is the significance of padding when training models? Explain in detail. *Medium*. https://medium.com/@sujathamudadla1213/what-is-the-significance-of-padding-when-training-models-explain-in-detail-3f257003ba8e

Pramoditha, R. (2023b, February 22). All you need to know about batch size, epochs and training steps in a neural network. *Medium*. https://medium.com/data-science-365/all-you-need-to-know-about-batch-size-epochs-and-training-steps-in-a-neural-network-f592e12cdb0a

PyTorch vs TensorFlow: Deep Learning Frameworks [2023]. (n.d.). https://www.knowledgehut.com/blog/data-science/pytorch-vs-tensorflow

RStudio. (n.d.). *Keras and TensorFlow Installation error · Issue #1311 · RStudio/Keras*. GitHub. https://github.com/rstudio/keras/issues/1311

Janse, B. (2023, May 17). *STARR Method for reflection explained*. Toolshero. https://www.toolshero.com/personal-development/starr-method/

Reddiculess. (n.d.). *New to ML, which is easier to learn - TensorFlow or PyTorch? : r/learnmachinelearning*.https://www.reddit.com/r/learnmachinelearning/comments/12p9bbt/new_to_ml_which_is_easier_to_learn_tensorflow_or/

Sharma, P. (2022, 22 maart). *A basic introduction to TensorFlow in deep learning*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/03/a-basic-introduction-to-tensorflow-in-deep-learning/

Sudhakar Sivanesan | LinkedIn. (n.d.). https://www.linkedin.com/in/sudhakar-sivanesan-21874526/

TensorFlow Core | Machine learning for beginners and experts. (n.d.). TensorFlow. https://www.tensorflow.org/overview

Team, K. (n.d.-c). Keras Documentation: Developer Guides. https://keras.io/guides/

Trouble installing keras. (n.d.). Stack Overflow. https://stackoverflow.com/questions/74640526/trouble-installing-keras

TensorBoard | TensorFlow. (n.d.). TensorFlow. https://www.tensorflow.org/tensorboard

Towards Data Science. (2001b, december 21). *Towards data science*. https://towardsdatascience.com/

Tensorflow. (n.d.). *Issues · Tensorflow/tensorflow*. GitHub. https://github.com/tensorflow/tensorflow/issues

Team, K. (n.d.). *Keras Documentation: About Keras 3*. https://keras.io/about/

"PIP" is not recognized as an internal or external command. (n.d.). Stack Overflow. https://stackoverflow.com/questions/23708898/pip-is-not-recognized-as-an-internal-or-external-command

Install TensorFlow 2. (n.d.). TensorFlow. https://www.tensorflow.org/installVivek. (n.d.). *How to install tensorflow please help: r/MLQuestions*. https://www.reddit.com/r/MLQuestions/comments/s6vnw0/how_to_install_tensorflow_please_help/

Codebasics. (n.d.). *Data Analytics Online Course with Certificate*. Codebasics. https://codebasics.io/

Median: How to Find it, Definition & Examples - Statistics How To. (2023, March 4). Statistics How To. https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/median/

Epochs, Batch Size, Iterations - How they are Important. (n.d.). https://www.sabrepc.com/blog/Deep-Learning-and-AI/Epochs-Batch-Size-Iterations

Installation — Anaconda documentation. (n.d.). https://docs.anaconda.com/free/anaconda/install/index.html

Terra, J. (2023, August 23). *Keras vs Tensorflow vs Pytorch: Key Differences Among Deep Learning*. Simplilearn.com. https://www.simplilearn.com/keras-vs-tensorflow-vs-pytorch-article

Mahto, P. (2021, December 15). PANDAS for Machine Learning - MLPoint - Medium. *Medium*. https://medium.com/mlpoint/pandas-for-machine-learning-53846bc9a98b

Klingler, N. (2023, March 10). *Jupyter Notebook for Machine Learning – A gentle introduction*. viso.ai. https://viso.ai/deep-learning/jupyter-notebook-for-machine-learning/

Brownlee, J. (2020, June 23). *Softmax Activation Function with Python*. MachineLearningMastery.com. https://machinelearningmastery.com/softmax-activation-function-with-python/

Should I repeat lengthy deep learning experiments to average results ? How to decide how many times to repeat? (n.d.). Artificial Intelligence Stack Exchange. https://ai.stackexchange.com/questions/31737/should-i-repeat-lengthy-deep-learning-experiments-to-average-results-how-to-de

Towards Data Science. (2001a, december 20). *Towards data science*. https://towardsdatascience.com/

Allgaier, J., Mulansky, L., Draelos, R. L., & Pryss, R. (2023). How does the model make predictions? A systematic literature review on the explainability power of machine learning in healthcare. *Artificial Intelligence in Medicine*, *143*, 102616. https://doi.org/10.1016/j.artmed.2023.102616

Brownlee, J. (2021, January 12). *Gentle introduction to the ADAM Optimization Algorithm for deep learning*. MachineLearningMastery.com. https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

eHowEducation. (2010, March 29). *Research papers : How to cite an interview in APA style* [Video]. YouTube. https://www.youtube.com/watch?v=GwrU7ku9k4c

AiOTA LABS. (2021, December 7). Quantifying accuracy and SoftMax prediction confidence for making safe and reliable deep neural network based AI system. *Medium*. https://medium.com/@aiotalabs/quantifying-accuracy-and-softmax-prediction-confidence-for-making-safe-and-reliable-deep-neural-6dbb8e360e15

Zulkifli, H. (2019, March 28). Understanding learning rates and how it improves performance in deep learning. *Medium*. https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10

Dey, V. (2021, November 20). *When to use One-Hot Encoding in deep Learning?* Analytics India Magazine. https://analyticsindiamag.com/when-to-use-one-hot-encoding-in-deep-learning/

About us. (n.d.). https://www.versuni.com/about-us/

Tucci, L., & Burns, E. (2023, September 15). *What is machine learning and how does it work? In-depth guide*. Enterprise AI. https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML

Vogel, J. (n.d.). *ICT Research Methods — Methods Pack for research in ICT*. ICT Research Methods. https://ictresearchmethods.nl/

Boog, J. (2024, January 8). *16 Best Machine learning Software reviewed for 2024*. The QA Lead. https://theqalead.com/tools/best-machine-learning-software/

De Philips 7000 Series Steelstofzuiger Aqua XC7057/01 Steelstofzuiger Aqua kopen. (n.d.). Philips. https://www.philips.nl/c-p/XC7057_01/7000-serie-snoerloze-stofzuiger-aqua

Appendix A

```python
# Load a list of files and return as a 3D numpy array
def load_group(filenames, path=''):
    loaded = list()
    shapes_before_padding = []

    for name in filenames:
        data = load_file(path + name)
        loaded.append(data)
        shapes_before_padding.append(data.shape)

    # Pad or truncate sequences to the length of the shortest sequence
    loaded = pad_sequences(loaded, padding='post', truncating='post', dtype='float32')

    # Print shapes before and after padding/truncating
    print(f"Shapes before padding/truncating: {shapes_before_padding}")
    print(f"Shapes after padding/truncating: {loaded.shape}")

    return loaded
```

Figure 1: This code snippet demonstrates a typical process of loading sequence data, followed by padding or truncating to ensure uniform length, which is crucial for training a deep learning model effectively.

```python
# Libraries that are required for deep learning model
import pandas as pd
import numpy as np
from numpy import mean
from numpy import std
from pandas import read_excel
from numpy import dstack
from pandas import read_csv
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers import Conv1D, Dropout, MaxPooling1D, Flatten, Dense, BatchNormalization
from keras.optimizers import Adam
from keras.utils import to_categorical
from keras.optimizers import RMSprop
from keras.layers import GlobalAveragePooling1D
from keras.preprocessing.sequence import pad_sequences
from sklearn.utils import class_weight
import tensorflow as tf
tf.keras.backend.set_floatx('float32')

print('hello') #this print is just for my personal satisfaction or confirmation that all th
```

[55] ✓ 0.0s

```
hello
```

+ Code + Markdown

```python
# Load a single file as a pandas DataFrame
def load_file(filepath, skiprows=1):
    dataframe = pd.read_excel(filepath, header=None, skiprows=skiprows)
    return dataframe.values
```

Figure 2: This image demonstrate that user can write code step by step into different column's and required libraries for the machine learning model.

```python
def evaluate_model(trainX, trainy, testX, testy, label_names):
    verbose, epochs, batch_size = 0, 10, 16  # put batch_size 32 when the data is huge
    print(f"Shape of trainX before reshaping: {trainX.shape}")

    # Reshape trainX if it's a 2D array
    if len(trainX.shape) == 2:
        trainX = trainX.reshape(trainX.shape[0], trainX.shape[1], 1)

    print(f"Shape of trainX after reshaping: {trainX.shape}")
    print(f"Shape of trainy: {trainy.shape}")

    # Assuming trainy is one-hot encoded, extract the number of classes
    n_outputs = trainy.shape[1] if len(trainy.shape) > 1 else 1

    model = Sequential()
    model.add(Conv1D(filters=64, kernel_size=2, activation='relu', input_shape=(trainX.shape[1], trainX.shape[2])))
    model.add(Conv1D(filters=64, kernel_size=2, activation='relu', padding='same'))
    model.add(Dropout(0.5))
    model.add(MaxPooling1D(pool_size=1))  # Reduce the pool size
    model.add(Flatten())
    model.add(Dense(100, activation='relu'))
    model.add(Dense(n_outputs, activation='softmax'))
    optimizer = Adam(learning_rate=0.001)
    model.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
#     model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    # Fit network and get training history remove it when not needed
    history = model.fit(trainX, trainy, epochs=epochs, batch_size=batch_size, verbose=verbose,
                        validation_data=(testX, testy))
    # Fit network
    model.fit(trainX, trainy, epochs=epochs, batch_size=batch_size, verbose=verbose)
    # Evaluate model
    _, accuracy = model.evaluate(testX, testy, batch_size=batch_size, verbose=0)# changed to 1 to print the training progress
```

Figure 3: Here you can see the functionality of all 3 functions that plays crucial role to increase model accuracy (for this case the default values used except 'batch_size' because we are testing model with different values if it make some difference).

python **Step 1**

```
!pip install tensorflow
!pip install keras
```

python **Step 2**

```python
import tensorflow as tf
import keras
print(f'TensorFlow version: {tf.__version__}')
print(f'Keras version: {keras.__version__}')
```

Figure 4: Steps to be taken for installing TensorFlow and keras.

```
Shapes before padding/truncating: [(30999, 2)]
Shapes after padding/truncating: (1, 30999, 2)
Shapes before padding/truncating: [(5149, 2)]
Shapes after padding/truncating: (1, 5149, 2)
Shape of trainX before reshaping: (30999, 2)
Shape of trainX after reshaping: (30999, 2, 1)
Shape of trainy: (30999, 2)
161/161 [==============================] - 0s 911us/step
Overall Predictions:
Label_1:Hard Floor: [0.5359572 0.5359572 0.5359572 ... 0.5359572 0.5359572 0.5359572]
Label_2:Soft Floor: [0.4640428 0.4640428 0.4640428 ... 0.4640428 0.4640428 0.4640428]
>#1: 50.592
Shape of trainX before reshaping: (30999, 2)
Shape of trainX after reshaping: (30999, 2, 1)
Shape of trainy: (30999, 2)
161/161 [==============================] - 0s 941us/step
Overall Predictions:
Label_1:Hard Floor: [0.5356969 0.5356969 0.5356969 ... 0.5356969 0.5356969 0.5356969]
Label_2:Soft Floor: [0.4643031 0.4643031 0.4643031 ... 0.4643031 0.4643031 0.4643031]
>#2: 50.554
Ground Truth after final test:
Hard Floor
[50.59235095977783, 50.55350661277771]
Accuracy: 50.573% (+/-0.019)
```

Figure 5: Results after a completion of 2 continued tests and final accuracy

```
optimizer = Adam(learning_rate=0.001)
model.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
# Fit network and get training history remove it when not needed
history = model.fit(trainX, trainy, epochs=epochs, batch_size=batch_size, verbose=verbose,
                    validation_data=(testX, testy))
# Fit network
model.fit(trainX, trainy, epochs=epochs, batch_size=batch_size, verbose=verbose)
```

Figure 6: Here, an instance of the Adam optimizer is created with a specified learning rate of 0.001. The learning rate determines the step size during the optimization process. The model is compiled, specifying the loss function ('binary_crossentropy' in this case, commonly used for binary classification tasks), the optimizer (Adam in this case), and the metric to be used for evaluation (accuracy in this case).

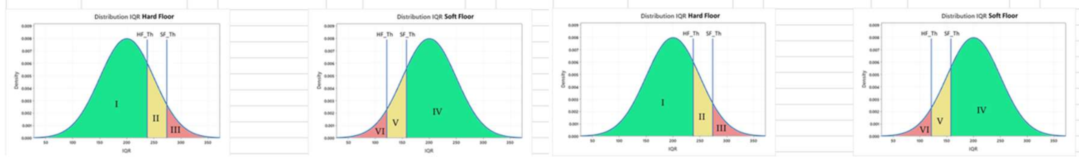
```
# Specify the learning rate
learning_rate = 0.0001  # You can adjust this value by 0.001
# Create an instance of the Adam optimizer with the specified learning rate
optimizer = RMSprop(learning_rate=learning_rate)
#Binary Classification:If you are dealing with a binary classification problem (i.e., you have
#only two classes), you should use 'binary_crossentropy' instead of 'categorical_crossentropy' as your loss function.
model.compile(loss='binary_crossentropy', optimizer=optimizer, metrics=['accuracy'])
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
# Fit network and get training history remove it when not needed
history = model.fit(trainX, trainy, epochs=epochs, batch_size=batch_size, verbose=verbose, validation_data=(testX, testy))
```

Figure 7: In the provided code snippet, the learning rate for the model's optimizer is specified as 0.0001. The learning rate is a hyperparameter that determines the step size at each iteration while updating the model weights during training. A smaller learning rate, such as 0.0001, means smaller steps, which can be beneficial for fine-tuning the model. The optimizer being used is RMSprop (Root Mean Square Propagation), which is a popular optimization algorithm.

Figure 8: Defined carpet types for testing.

Result on FTR (comparison with new)

	FTR prediction model New Nozzles								FTR prediction model Scrub test Nozzles							
	Foor recognized		Floor recognized incl. "no-change zone"		Floor wrong determined		Floor Type		Foor recognized		Floor recognized incl. "no-change zone"		Floor wrong determined		Floor Type
Home Lab	95.619	I	99.995	I+II	0.005	III	HF	Home Lab	99.999	I	100.000	I+II	0.000	III	HF
Rough Tiles	80.193	I	99.888	I+II	0.112	III	HF	Rough Tiles	99.500	I	99.999	I+II	0.001	III	HF
Wilton	56.700	IV	91.351	IV+V	8.649	VI	SF	Wilton	2.500	IV	37.000	IV+V	63.000	VI	SF
MediaMarkt	86.181	IV	99.908	IV+V	0.092	VI	SF	MediaMarkt	0.010	IV	50.000	IV+V	50.000	VI	SF
Loop Pole	98.729	IV	99.927	IV+V	0.073	VI	SF	Loop Pole		IV		IV+V		VI	SF



Conclusion:
After putting the nozzles in the scrub test, on all floors the IQR drops significantly (See ANOVA's).
As a result the hard floors are always detected as hard floors, but the tested soft floors are almost always detected as well as hard floors.
It must be noted that the result is only valid for a Wilton and MediaMarkt floor, as the more common Loop Pole (blue) floor is not available anymore.

Figure 9: These test results were done by employing a vacuum cleaner on different floor types. The tests on the left side utilized a brand-new brush in the nozzle, while those on the right side used an older brush. It's essential to note that these tests were conducted using the current algorithm, and no machine learning model was used in the process.

```
Shapes before padding/truncating: [(30999, 2)]
Shapes after padding/truncating: (1, 30999, 2)
Shapes before padding/truncating: [(5149, 2)]
Shapes after padding/truncating: (1, 5149, 2)
161/161 [==============================] - 0s 2ms/step
Overall Predictions:
Label_0: Hard Floor: [0.5757942  0.5838298  0.42848632 ... 0.61913615 0.6903392  0.6903392 ]
Label_1: Soft Floor: [0.42420584 0.4161702  0.57151365 ... 0.38086376 0.3096608  0.3096608 ]
>#1: 78.967
161/161 [==============================] - 0s 1ms/step
Overall Predictions:
Label_0: Hard Floor: [0.48011637 0.48862362 0.29663455 ... 0.33397272 0.62901914 0.62901914]
Label_1: Soft Floor: [0.51988363 0.51137644 0.70336545 ... 0.66602725 0.37098092 0.37098092]
>#2: 75.937
Ground Truth after final test:
Hard Floor
[78.96679043769836, 75.9370744228363]
Accuracy: 77.452% (+/-1.515)
```

Figure 10: These results came after first 40 median values were subtracted in the dataset and reducing the learning rate to 0.0001.

```python
# Load the dataset
file_path = r"C:/model testing/old and latest data 31k samples both/individual datasets/Soft floor samples test.xlsx"
df = pd.read_excel(file_path)

# Assuming your dataset columns are named 'timestamp' and 'Motor data'
motor_current_column = 'Motor data'

# Calculate the median of the first 40 elements in the 'Motor data' column
median_motor_current = df[motor_current_column].head(40).median()

# Subtract the median from the 'Motor data' column
df['Motor data'] = df['Motor data'] - median_motor_current

# Save the modified dataset to the original file
df.to_excel(file_path, index=False)

# Print the modified dataset
print("Modified Dataset:")
print(df)
```

0.9s

```
Modified Dataset:
      Time stamp  Motor data
0          13402       -77.5
1          13452        44.5
2          13502       202.5
```

Figure 11: This is the code that was used to subtract first 40 samples from the motor data column in the dataset for both training and testing.

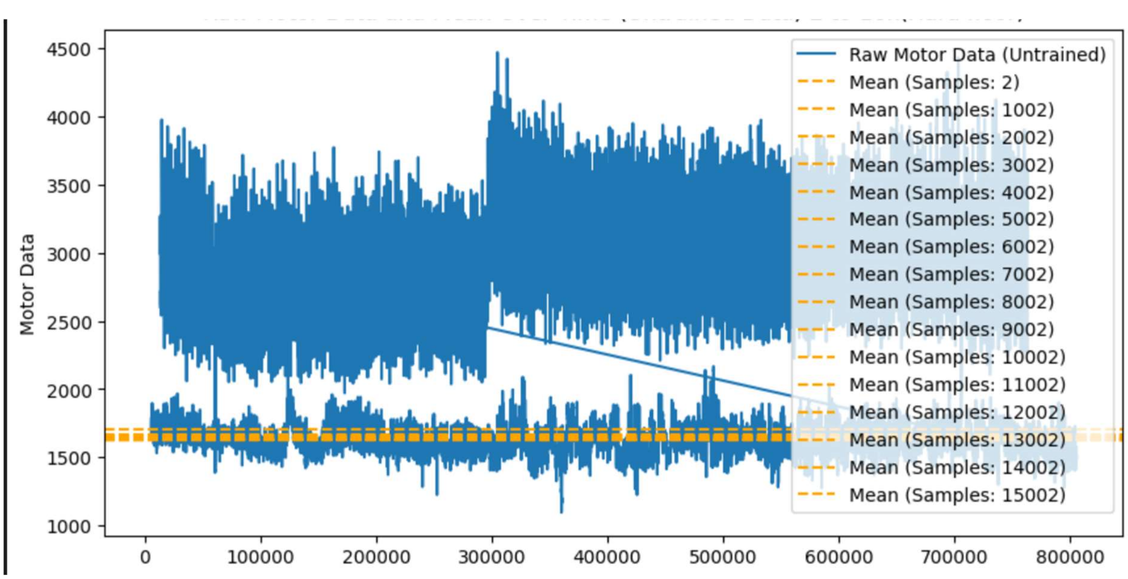


Figure 12: The data first printed on a chart to see the motor current fluctuations. The below with yellow line is motor current and the above variations are time stamps.

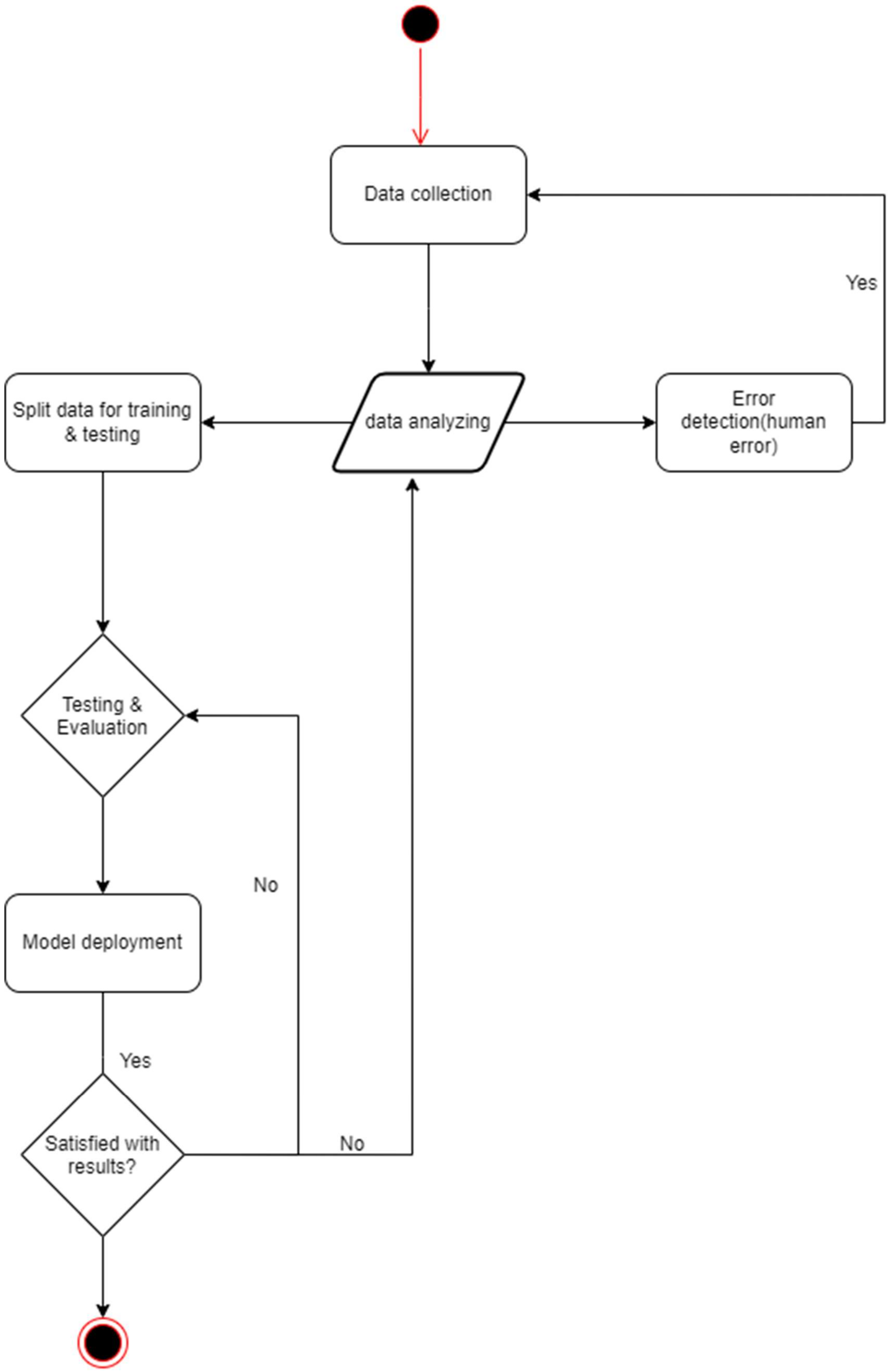


Figure 13: Overview of how machine learning process works.

Appendix B

The interviews with Sudhakar Sivanesan Machine learning expert:

Q: What is the best tool for coding when working on machine learning models?

Interview talks:

Paras: Hey Sudhakar! I'm going to start machine learning and wondering if I should use Jupyter or VS Code. Any thoughts?

Sudhakar: Hi Paras! Good question. Jupyter is like a smart notebook. It's great for playing with data and seeing results step by step. VS Code, on the other hand, is more like an all-in-one toolbox for coding, including machine learning.

Paras: Got it. So, Jupyter is like a data engineers' tool. What about VS Code?

Sudhakar: Exactly. VS Code is a bit more like a coding workshop. It's capable and can handle lots of different tasks. Great for building bigger projects.

Paras: Cool. But I heard Jupyter is better for using machine learning libraries.

Sudhakar: Yes, Jupyter is super-friendly with libraries. But here's the thing – VS Code has caught up. With some add-ons, it can do almost the same tricks.

Paras: Nice! I also need to collaborate with others. Jupyter seems good for that, right.

Sudhakar: Absolutely. Jupyter makes it easy to share your work.

Paras: That's handy. But I'm worried about learning all this. Which one is easier for beginners?

Sudhakar: Jupyter is often the starting point. It's like learning to ride a bike with training wheels. But, hey, don't be scared of VS Code. Once you get used to it, it's a powerful tool.

Paras: Okay, I'll keep that in mind. So, project-wise, what would you recommend?

Sudhakar: If you're playing with data a lot, start with Jupyter. For more serious coding and bigger projects, give VS Code a shot.

Paras: Thanks a bunch, Sudhakar! I'll figure out which one suits my project best.

Sudhakar: No problem, Paras! Tell me if you have any questions.

Q: What machine learning extensions and libraries are considered suitable for this project, and how were they chosen?

Interview talks:

Paras: Hi Sudhakar, any advice on which machine learning extensions or libraries I should consider?

Sudhakar: For starters, I would recommend Keras and TensorFlow. They're like the dynamic duo in the deep learning world.

Paras: Ah, heard about them. What's the deal with Keras and TensorFlow?

Sudhakar: Think of TensorFlow as the engine running the show. It's super powerful and efficient for heavy lifting in deep learning. Keras, on the other hand, is like the friendly face that makes working with TensorFlow easier. It's a high-level interface.

Paras: So, they're a team or together?

Sudhakar: Exactly! Keras is now integrated into TensorFlow. You can use them separately, but for your project they will be used together.

Paras: Cool. But what makes them good for deep learning?

Sudhakar: TensorFlow has the muscle for complex computations and handles things like neural network layers. Keras adds simplicity. You can design and train deep learning models with just a few lines of code.

Paras: Sounds user-friendly. Are they good for beginners?

Sudhakar: Absolutely. Keras, especially, is beginner friendly. It abstracts some complexities, making it easier to grasp concepts. As you get more comfortable, you can dive into TensorFlow's deeper.

Paras: Nice. And for a deep learning project, are they the go-to choice?

Sudhakar: In many cases, yes. They're widely used in the deep learning community and also in industries. Plus, there's a ton of documentation and community support.

Paras: Got it. Any challenges I should be aware of?

Sudhakar: Well, deep learning itself can be challenging. But specifically, with Keras and TensorFlow, keeping up with updates can be crucial. Some functions might change, and you want to be using the latest and greatest.

Paras: Solid advice. I'll keep an eye on updates. Thanks, Sudhakar!

Sudhakar: No problem, Paras! Enjoy the deep learning with Keras and TensorFlow and we can have our next meeting on Monday.

Q: What are the reasons for opting for a deep learning model? and not classical machine learning?

Interview talks:

Paras: Hi Sudhakar! I've been wondering, why should we go for a deep learning model instead of classical machine learning for our project?

Sudhakar: Hey Paras! Good question. Deep learning shines when dealing with complex patterns and huge datasets. It's like teaching the computer to learn on its own, finding those intricate details.

Paras: But isn't classical machine learning doing the job too?

Sudhakar: Absolutely, for many tasks it can. But deep learning thrives when there's lots of data, especially if it has patterns that might be tricky for traditional methods to catch.

Paras: Interesting. Any specific reasons you'd lean towards deep learning?

Sudhakar: Well, deep learning is fantastic with things like image and speech recognition. If we're dealing with time series signals, which we are, deep learning can unfold hidden patterns better than classical methods.

Paras: Got it. So, it's about the kind of data we're working with?

Sudhakar: Exactly! Deep learning is like having a super detailed algorithm in its model for certain types of data – it can dig deeper into patterns that classical methods might miss.

Paras: Okay sounds convincing. Any drawbacks?

Sudhakar: Well, deep learning can be hungry for data and resources. If we have a solid amount of data, which we do, and the right setup, deep learning is often the winner.

Paras: Okay, let's go with deep learning then, especially with our time series signals data. Thanks, Sudhakar!

Sudhakar: No problem, Paras! Deep learning is like having a powerful tool in our toolkit, and for our project, it's the right fit. See you then in the next meeting.

Q: Why opt for the 'verbose,' 'epochs,' and 'batch_size' functions in my deep learning ML model?

Interview talks:

Paras: Hi Sudhakar! I'm exploring some function in our model and wondering why we need to bother with things like 'verbose,' 'epochs,' and 'batch_size.' Do they really matter?

Sudhakar: Hey Paras! Absolutely, they're the important parameter or functions that help our model to learn better. Let me break it down. 'Verbose' is like choosing how much the model talks during training. You can have it silent, chatty, or something in between.

Paras: Okay, got it. What about 'epochs'?

Sudhakar: 'Epochs' are like how many times our model goes through the entire training data. Too few, and it might not learn enough; too many, and it might memorize instead of understanding.

Paras: Ah, okay!

Sudhakar: And 'batch_size' is like how much data our model handles at once. Big batches might be too much for the computer, small ones might not capture the whole picture. It's about balance.

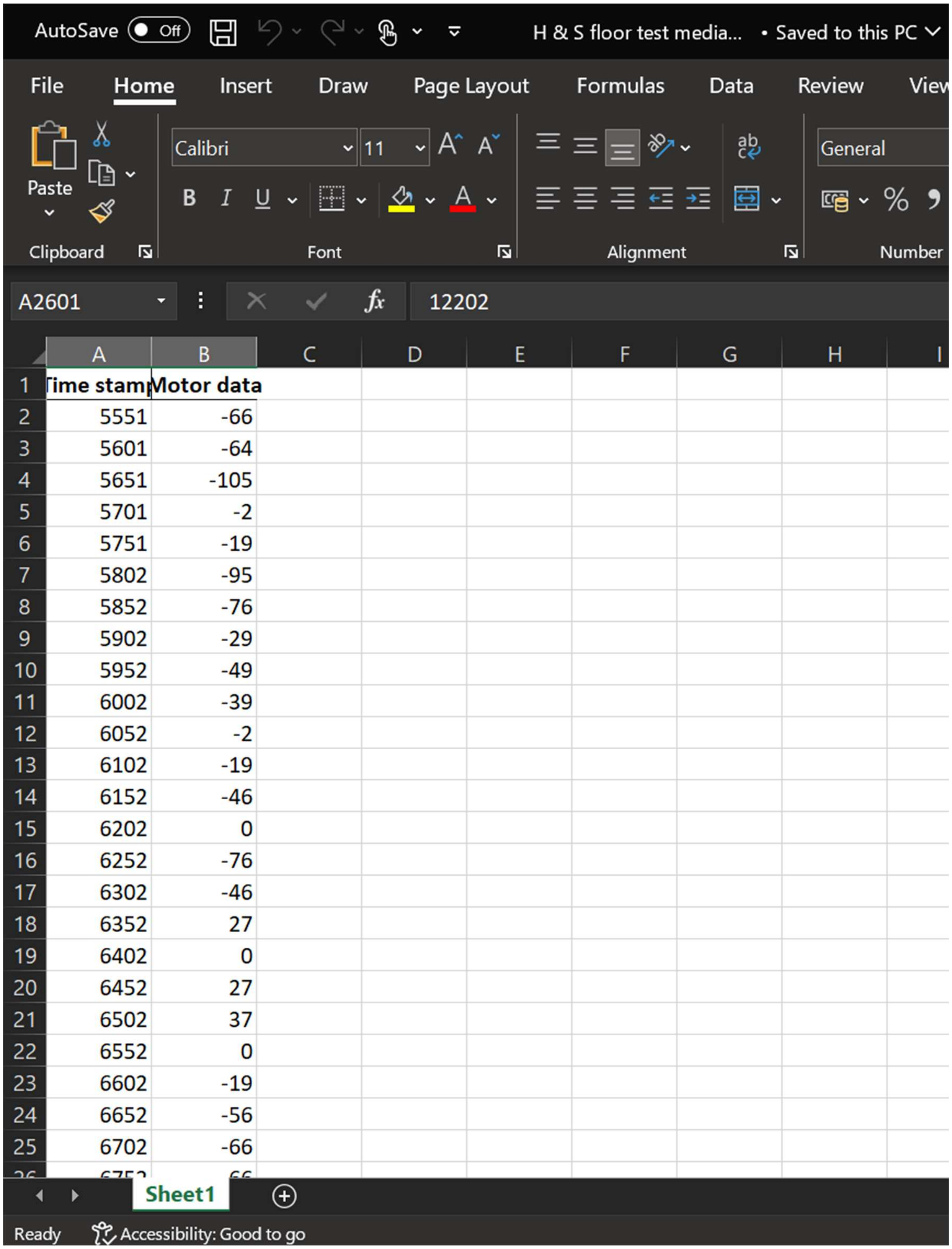
Paras: So, it's like tuning our model.

Sudhakar: Yes! You want your model to play the perfect tune, not too fast, not too slow, and not missing any beats.

Paras: Perfect! I'll make sure our model learns correctly while training by setting up right values using these parameters.

Sudhakar: Okay great then! Tweaking these settings can make a huge difference in how well our model learns and performs.

Appendix C



	A	B
1	Time stamp	Motor data
2	5551	-66
3	5601	-64
4	5651	-105
5	5701	-2
6	5751	-19
7	5802	-95
8	5852	-76
9	5902	-29
10	5952	-49
11	6002	-39
12	6052	-2
13	6102	-19
14	6152	-46
15	6202	0
16	6252	-76
17	6302	-46
18	6352	27
19	6402	0
20	6452	27
21	6502	37
22	6552	0
23	6602	-19
24	6652	-56
25	6702	-66

Figure 14: This is how the motor cureent data looked like after logged from the vacuum cleaner via Bluetooth module.

Appendix D

Project plan.pdf