Master's thesis

Minimize experimentation overhead through dataset selection, ensemble feature attention, and feature selection with reduced subset sizes

Mihai Anton

Student Name	Student Number
Mihai Anton	5350123

Thesis Advisor:	Arie van Deursen
Daily Supervisor:	Luis Miranda da Cruz
Daily Co-Supervisor:	Arumoy Shome
Google Supervisors:	Vincent Cohen-Addad & Sammy Jerome
Project Duration:	December, 2022 - October, 2023
Faculty:	Electrical Engineering, Mathematics & Computer Science

TU Delft

Summary

In large-scale ML, data size becomes a critical variable, especially in the context of large companies, where models already exist and are hard to change and fine-tune. Time to market and model quality are essential metrics, thus looking for ways to select, prune and augment the input data while treating the model as a black box can speed up the process from raw data to productionized model.

Datasets can have thousands of features and many redundant/duplicate samples, for various business logic reasons. In some particular ML flows, it might be that only a subset of them provide most of the input to the final accuracy. Also, looking into ways to provide insights on what data points are the most meaningful can help engineers collect more relevant samples, or focus their attention on specific parts of the data distribution.

Contents

Summary i

1 Introduction 1

2 Research Questions and Problems 4
2.1 Dataset Pruning . . . 4
2.2 Using fast proxy models . . . 4
2.3 Data redundancy and collection feedback . . . 4

3 Background 6
3.1 Literature Survey . . . 6
3.1.1 Data Quality and Dataset Selection . . . 6
3.1.2 Approximated Pipeline Execution using Proxy Models . . . 8
3.1.3 Data Collection Feedback . . . 10
3.2 The Enterprise Machine Learning process . . . 12

4 Systematic Literature Review 15
4.1 SLR overview . . . 15
4.2 Methodology . . . 15
4.2.1 Search query and search engine . . . 15
4.2.2 Inclusion/exclusion criteria . . . 16
4.2.3 Snowballing . . . 18
4.3 Paper Attributes and Questions . . . 18
4.3.1 Attributes . . . 18
4.3.2 Questions . . . 21
4.3.3 Notes format for each paper . . . 22
4.4 Final Paper List . . . 23
4.5 Preliminary Insights . . . 23
4.6 Research types . . . 25
4.7 Time Reduction Strategy . . . 30
4.8 Feature Selection Approaches . . . 32
4.9 Datapoint Selection Approaches . . . 36
4.10 Domains . . . 37
4.11 Secondary publications . . . 40
4.12 Common Patterns in Data Selection . . . 41

5 Advancing feature selection performance at corporate scale 43
5.1 Objective . . . 43
5.1.1 Sequential Attention . . . 43
5.1.2 Improving Machine Learning Iterations . . . 44
5.2 Methodology . . . 45
5.2.1 Research directions considered . . . 46

5.3 Methods and Experiments 48
5.3.1 Datasets setup 48
5.3.2 Improving the Sequential attention subset size 49
5.3.3 Ensemble sequential attention 52
5.3.4 Sequential attention with feature batches 54
5.3.5 Intermediate Sequential Attention 55
5.3.6 Sequential attention on reduced data 57
5.3.7 Computing mask size upper bound with SVD 61
5.3.8 End-to-end time comparisons 62
5.4 Feature Masking Tool 65
5.4.1 Input 65
5.4.2 Output 65
5.4.3 Architecture 66
5.4.4 Usage 66
5.5 Results and Impact 67
5.5.1 Training time saving 67
5.5.2 Reduced datasets 68
5.5.3 Reduced inference time 68
5.5.4 New research directions 68
5.5.5 Code 68

6 Conclusion 69

References 71

1 Introduction

As the research in AI and data blossomed in the 21st century, by 2023 data is the new gold and dictates most of our decisions. This can come in many shapes and variants, from autonomous cars to language models that accurately answer questions with relevant information [33]. Data collection has a pace like never before [20], and the need to transform it into predictive models and insightful analyses is more and more relevant. Big companies like Google, Meta, Microsoft, and Apple produce enormous amounts of data [18] [20], together with a multitude of startups, both in the data industry, as well as in IoT, Crypto, or Fintech. Most of the companies take data driven decisions, from what projects to prioritize internally, to what products to recommend and what ads to show to the end user. The need for data driven product decisions is so high that the quality of the data is becoming of paramount importance [11].

Data quality is extremely important in today's data-driven world. It affects everything from business decisions and strategic planning to customer satisfaction and regulatory compliance. Poor quality data can lead to incorrect conclusions, inefficient processes, loss of revenue and missed opportunities. On the other hand, high-quality data is reliable, accurate, and relevant, which enables organizations to make informed decisions, improve operations, and drive innovation.

Ensuring data quality is a continuous process that involves identifying, correcting, and preventing errors in data. It requires having proper data governance and management policies in place, as well as implementing data quality checks and controls. Organizations need to invest in data quality tools and technologies to automate and streamline these processes, and they need to train their employees on how to handle and use data responsibly. By prioritizing data quality, organizations can maximize the value of their data assets and gain a competitive advantage in their industry. On top of this, in the context of machine learning, automatic data selection and data engineering steps need to be put in place so that the systems are able to learn and improve by themselves, without human intervention, which is time consuming.

On the other hand, continuously collecting data can lead to enormous databases and overloaded warehouses. An online study shows that in 2021, humans were generating 1134 Trillion MBs of data on a daily basis, and this quantity is ever-growing with the advancements in the AI and IoT fields. But this data alone does not provide much business value, unless analyzed and transformed into useful modeling tools, such as Machine Learning models, making predictions and serving users worldwide. With this mention, data quality is also an important metric to quantify. Quantity would

be nothing without quality, and the more the data grows, the more there is a need to filter it out and aim to extract the signal from the noise.

In recent years, machine learning has become an increasingly important tool for analyzing and predicting outcomes based on data. However, the success of machine learning models depends on the quality and relevance of the data they are trained on. Therefore, the process of selecting and preparing data for machine learning is crucial for obtaining accurate and reliable results. Also, in the process of creating a high-quality machine learning model, there is a lot of experimentation involved, which translates to a large number of computing hours that often don't have a direct correlation to the quality of the final result.

In this thesis, our goal is to minimize the experimentation overhead, while maximizing the quality of the result. The main optimizations we plan to look into are dataset selection (removing data in a consistent and replicable manner, so useless information does not slow down the process) and using proxy models to help prune the dataset (using a fast learner instead of a more robust neural network that takes more time to train). We use the classification niche since it is a very common scenario (for both simple datasets, as well as more complex usecases such as image recognition and speech detection) and it's also highly structured, both in the shape of the data and in the expected output of the model. Further, we discuss the importance of data selection in machine learning and the various factors that need to be considered when choosing data for training and testing models. We also explore the various techniques and best practices that can be used to improve the quality and suitability of data for machine learning purposes. By understanding the importance of data selection and taking a careful and systematic approach to selecting and preparing data, organizations can ensure that their machine-learning models are able to deliver the most accurate and valuable insights.

The thesis is structured as follows: in Chapter 2, we formalize the goals and problems we have to overcome for each research area. Then, in Chapter 3 we do a literature survey of related work on the topics we want to handle, summarising papers and internet publications on the subject at hand. In Chapter 4 we do an in depth and replicable survey of the recent literature on the data pruning and feature selection fiels. Then, in Section 5 we advance the state of the art in feature selection by finding trade-offs between feature selection performance and total runtime. Then, in Chapter 6 we conclude and summarise the main findings of the thesis.

The work in this thesis presents significant contributions to the field of large-scale machine learning, particularly in minimizing experimental overhead through innovative dataset selection, ensemble feature attention, and finding trade-offs between feature selection algorithm runtime and performance. Key results include the development of efficient dataset pruning techniques, the effective use of sequential attention, and insights into reducing data redundancy. These findings not only demonstrate significant advancements in reducing the time and computational resources required for machine learning iterations but also enhance the quality and interpretability of resulting models.

The implications of this research are profound, particularly for data-driven companies like Google, who stand to benefit from these optimized approaches. The methodologies and insights presented here pave the way for more efficient, data-driven decision-making processes, marking a notable contribution to the ever-evolving landscape of artificial intelligence and data analysis. This thesis aims to provide an incremental approach to finding better feature selection solutions, guided by experiments and state-of-the-art solutions, offering a comprehensive understanding of the research journey and its substantial impact on the field.

2
Research Questions and Problems

This chapter introduces the main research questions of our study, along with the potential methods we consider for answering them. Each section focuses on a different question, from improving dataset pruning to exploring the use of fast proxy models, and examining the impact of up-weighting underrepresented classes. We also look into reducing data redundancy and enhancing data collection methods. These questions are the driving force behind our research, and this chapter lays out the initial steps we take to address them. Through these sections, you'll see the challenges we face and the innovative approaches we're exploring in machine learning.

2.1. Dataset Pruning

RQ1: What is an efficient way to prune a dataset that minimizes the time it takes to train the model as a black box, while maximizing the accuracy of the output, in relation to the same accuracy that would result from training the model with the full dataset?

- **Approach 1:** Find a way to remove features that would not reduce the quality.
- **Approach 2:** Find a way to remove/cluster samples that would not reduce the quality.

2.2. Using fast proxy models

RQ2: Can the use of a fast learning proxy model help with deciding what data selection mechanisms to apply? What is the best way to minimize the experimentation overhead by approximating the full pipeline with a simpler model and a subset of the full dataset?

- **Approach 3:** Find a model that trains fast and is able to be a proxy for the more complex model (random forest for example).

2.3. Data redundancy and collection feedback

RQ3: What are the common data redundancy issues in datasets? How can an initial experiment on the dataset provide insights into what data is the most representative and what samples should be removed/ what data points should be additionally collected?

- **Approach 4:** Find what classes to collect data from, that would increase the quality of the model.
- **Approach 5:** Find what classes are redundant in the dataset, whose removal keeps the model quality the same.

Data pruning and dataset selection are important techniques in the field of machine learning, as they can help improve the performance and efficiency of a model. One approach to data pruning is to remove irrelevant or redundant data from the dataset. This can be done by using feature selection algorithms, which identify the most informative features of the data and remove the rest. This can be useful for reducing the dimensionality of the data, which can improve the performance of the model and decrease the amount of data that needs to be processed, and the training time required until convergence. However, this approach also has the potential to remove important features that are necessary for the model to make accurate predictions, leading to a model that might not have the ability to generalize.

Another approach to data pruning is to remove outliers from the dataset. Outliers are data points that are significantly different from the rest of the data and can have a negative impact on the performance of the model. This can be done by using outlier detection algorithms, which identify data points that are outside of a specified range or distribution. Removing outliers can improve the performance of the model, but, as before, can hinder the ability of the model to be flexible and generalize.

A third approach is to just use a subset of the data for training and testing. One common way of selecting a subset of data is to use a random sampling method, where data points are chosen at random from the dataset. This approach is quick and easy to implement, but it has the potential to select a biased subset of data that does not accurately represent the entire dataset. Another approach is to use stratified sampling, where data points are chosen to ensure that the proportion of different classes in the subset is similar to the proportion in the entire dataset. This approach can help reduce the bias of the subset of data, but it also has the potential to remove important data points that are necessary for the model to make accurate predictions.

This being said, the goal is to reduce the dataset size, both in terms of data points and number of features, but doing so in a way that still allows the model to generalize and learn from meaningful samples. We can look into 2 main different types of pruning: depth (drop samples) and width (drop features). For the first one, we can take K random subsets of the original dataset, keeping the distribution unchanged, and training a fast AutoML model on each. Given the results, we can decide what classes can be downsampled to reduce dataset size, and what classes should be upsampled to increase accuracy. The second type of pruning comes as a result of the analysis of the first. Looking into the activation of each feature, we can decide which features provide the most input to the model and what features should be dropped in the learning process.

3
Background

3.1. Literature Survey

As the general topics of data quality and machine learning have been on the rise in the last decade, we decided to start our research with a comprehensive literature survey, to understand the current state of the art when it comes to reducing the size of the data that has to be modeled while keeping the quality of the model high. We looked into papers on feature selection, dataset pruning, machine learning experimentation time minimization, and data collection feedback.

3.1.1. Data Quality and Dataset Selection

Some pieces of recent research on data quality point out more or less the same problems in data [19]. Those are data incompleteness, inconsistency, inaccuracy, duplication, and data points that are not up to date. Also, various research papers point out that data quality issues can come from both data collection (wrongly collected, inaccurate human input, not calibrated sensors, etc) and from the data domain itself (representation spaces that are hard to model, lots of redundant variables, etc). Our work will focus mostly on the second, namely, how can we deal with existing data, extract the signal from the noise, and reduce the size of the dataset that needs to be modeled as a consequence.

Also, [19] points out that companies underestimate the problem of data quality, and this fires back later, the consequences ranging from ”significant to catastrophic”, with results in failing projects, loss of revenue, and increased customer churn. This is especially true for businesses where data is a major player in the business model, where customer products are built on the data acquired. Once more, the issue of data quality in data-driven businesses is worth looking into. Furthermore, the same paper points out that the data in a business scenario might be very different than toy datasets used in research, which might have a common distribution and potentially known optimizations. This adds on the highly volatile field of data quality, which can change very frequently due to the rapidly moving AI/Data startup ecosystem.

They also mention using principal component analysis as a feature selection technique. Principal component analysis (PCA) is a statistical technique used to reduce the dimensionality of a dataset. It does this by finding a new set of linearly uncorrelated variables, called principal components, which capture the most important patterns in the data. These principal components are a linear combination of the original variables and are ranked in order of importance based on the amount of variation they

explain in the data [2]. They also mention Exploratory Factor Analysis (EFA), and state that it's equivalent to PCA in terms of the outcome, but they are different in how they accomplish their functions.

"The 10% You Don't Need" [9] offers an interesting insight into how to reduce the depth of the dataset while keeping the same dimension. They use Agglomerative Clustering, looking at a subsample of a dataset that performs better, rather than a subsample of the features. They set a predefined target (for example 90% of the initial dataset) and run the clustering algorithm until the target is reached. They ran an experiment on ImageNet, and the finding was that by removing 10% of similar samples, the ability of the model to generalize increased. The base of their research is papers that look into random dataset sampling, and iterative removal of small batches of the dataset while looking into the performance gains of the resulting model. While those approaches might work, the goal is to find a method that also minimizes the experimentation time, while keeping the dataset small and relevant. Looking specifically at their ImageNet study, they use the last layer of the model trained on the whole dataset as the embedding for Agglomerative Clustering. Although a good step in the process of finding the minimal relevant dataset, there is still a need of training the model on the whole dataset: "To find redundancies in datasets, we look at the semantic space of a pre-trained model trained on the full dataset.". Although a good point for research, one might ask about how this is relevant in a real, high-volume scenario, where researchers and engineers try to avoid using the whole dataset for training. Removing the 10% You don't need is an important improvement to a Machine Learning flow, and one question worth answering is how can this be achieved by avoiding training on the whole dataset during the process.

In their paper, Taleb et at. [43] describe a big-data generic framework for continuous quality management, and look into best practices from the inception of the system and continuously through development. They define 4 steps in the data collection pipeline: Generation, Acquisition, Storage, and Analysis. While analysis is something we are planning to touch on through the research questions above, our thesis aims to add a 5th step to this list: Data Enhancement. This can mean both feature engineering and data selection. Also, they define the big data 3V characteristics: Volume, Variety, and Velocity. In our approach, we plan to create a robust method for enhancing datasets keeping in mind all 3 dimensions. Volume makes the data hard to analyze at once, Variety makes it hard for a one size fits all solution to cover all edge cases, and Velocity makes it hard for comprehensive algorithms to keep up with the changes in the data.

In a study at Meta, Sorscher et al. [41] worked on developing a data pruning algorithm, that is based on a self-supervised metric, and argues that finding new data pruning metrics might provide a way to better neural scaling laws. They emphasize that nowadays, in order to change the accuracy metric of huge models even with just one percentage, the amount of data required is significant, and the higher you want the metric to go, the more data you need to feed in the model, which only makes the problem worse, since, as they state, many training examples are highly redundant.

One of the initial findings of their paper is if the initial data is abundant, only the hard examples need to be kept, while if the initial data is scarce, only the easy examples should be kept. One of the studies they refer to, [32], trains small ensembles, and decides whether a training sample is easy or hard by looking at the error for each data point in the shallow trained ensembles. Another study, [45], measures how easy (or hard) a model forgets some samples during the learning flow, and uses the metric in the data pruning step.

The study keeps emphasizing the link between the best pruning strategy and the amount of initial data. In this thesis, the main focus is providing an efficient data pruning strategy at a large scale, especially for enterprise and open source datasets. They also point out that classical randomly selected data performs poorly, especially because random data samples provide no additional information to the model, thus being redundant.

Sorscher et al. [41] also touch upon data pruning in the context of pre-trained models, mentioning that when it comes to big image/language models, having the model pretrained on a benchmark dataset (CIFAR-10, ImageNet) allows for more intensive data pruning in a transfer learning scenario. Also, they mention that ImageNet might be an isolated case when it comes to the amount of data that can be pruned, since the images inside have already been highly curated. In the context of the thesis, transfer learning is a common topic, since big companies don't train models from scratch on new data. Thus, a strategy for progressively pruning more and more data with each model training can be discussed. Given that the new data the model is being adjusted on is expected to have a similar distribution, using only a small representative subset can speed up the process.

3.1.2. Approximated Pipeline Execution using Proxy Models

Speed is an important factor in the training of machine learning models. The faster a model can be trained, the more quickly it can be deployed and put to use in real-world applications. In addition, faster training times also allow for more iterations and experimentation with different model architectures, which can lead to better performance and more accurate predictions. This is particularly important for large-scale projects, where the amount of data being processed can be significant and traditional training methods may be too slow to be practical.

Another important aspect of speed in machine learning model training is the ability to quickly adapt to new data and changing conditions. In many real-world applications, the data being used for training and prediction is constantly evolving, and models need to be able to adapt to these changes in order to remain accurate and effective. Faster training times make it possible to retrain models more frequently, which can help ensure that they are always up-to-date and able to make accurate predictions based on the latest data. This is crucial in fields such as finance, healthcare, and transportation, where the ability to quickly respond to changing conditions can have a significant impact on overall performance.

In their paper where they look into diagnosing network performance issues in client devices, Widanapathirana et al. [50] look into different machine learning model types, and use the following image 3.1 to draw conclusions on what model is the most efficient. Based on it, there is a substantial difference between using neural networks and using decision trees, especially when it comes to the speed of learning. In our thesis, we look into the usage of proxy models like decision trees to create an alternative machine learning pipeline, that would allow us to experiment with different data selection/transformation methods, without paying the price of a slow neural network. For the purpose of the thesis, we can easily conclude that SVMs and Neural Networks are the best when it comes to prediction quality, while Decision Trees are the best when it comes to speed of experimentation. Given this, Decision Trees/Random Forests might be a good alternative as a proxy model for quick experimentations on various data selection techniques.

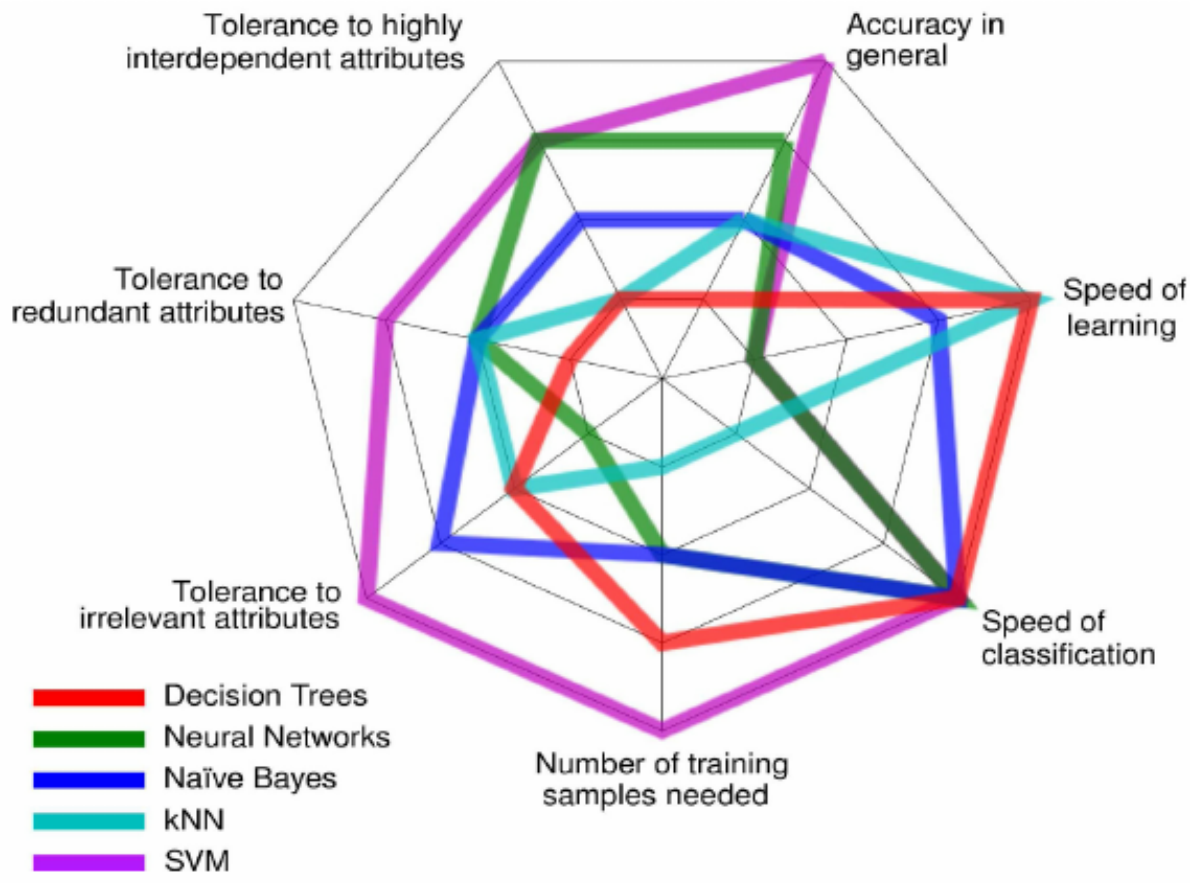

Figure 3.1: Comparison of Machine Learning Algorithms

In their paper, Paul et al [32] have the same goal as with the rest of the previous studies, removing as many redundant data points from the dataset, without observing drops in accuracy or model quality. They do so by using information from the model itself, after a brief period of training (enough to have relevant metrics, but not long enough to be equivalent to training the model on the full dataset). Although this is not an example of using a proxy model, a similar outcome can be reached by limiting the training time on a complex model.

They also stress the fact that more and more in the last decade people have tried to increase machine learning model accuracies by using overparametrized models and ever larger datasets. Trying to overcome this and lower experimentation time, they use the information in gradient norms after a few epochs of training on the same model as the final one. This metric helps them in ordering the samples by importance and difficulty and iteratively removing less important samples from the dataset. They find out that very early in training, redundant data can be identified by partial forgetting scores (how often a sample is learned and then forgotten). Based on that, samples that are easily forgotten in a few epochs are labeled as ”hard”, while sample that the model still classifies correctly as deemed ”simple”.

They run experiments with multiple pruning techniques over multiple datasets. Empirically, looking at the L2 norm of each sample early in the training flow and using forget scores performs the best on datasets like CIFAR10, CINIC10 and CIFAR100, showing that even removing 40% of some datasets does not drop final accuracy compared to a model trained on the full set of samples. In the context of the thesis, this work can be taken one step further, by combining forgetting scores with fast proxy models. Having this information, one can proceed and use only a subset of the samples, ordered by importance, in order to decrease the training load.

In their paper, Coleman et al. [14] look exactly into this. They use selection via proxy, which means using a less complex model for the data selection, and only train the actual model on the selected dataset. They explore this heuristic in the context of actively learning a representative dataset, and in core-set selection (like an algebraic basis but for a large dataset). They achieve up to 7% improvements in runtime by using a shallower model for ImageNet, and a 41.0% speed-up on the Amazon Review Polarity dataset. The authors also point out that significant amounts of time are spent on improving the model with a relatively small percentage.

3.1.3. Data Collection Feedback

Feedback loops are a very important element in machine learning flows. When it comes to large scale machine learning, using the feedback to improve the product is essential since in the real world many variables can change fast, from shifting distributions to changes in consumer behavior. However, as much as improving upon feedback is necessary, the source of the feedback might turn out to have either positive and negative outcomes for the machine learning model. In this chapter we’re looking into common feedback loop approaches for large scale machine learning, understand the pros and cons, and highlight how the thesis aims to improve the flows with the goal of outputting qualitative feedback on data collection.

The first type of feedback loop comes from the model itself. This refers to a process of continuous improvement in a model based on the output generated from previous inputs. This process involves using the model to make predictions on data, comparing the predicted results with actual results, and updating the model based on discrepancies. The cycle of prediction, evaluation, and improvement is repeated until the model

reaches a satisfactory level of accuracy. Feedback loops in machine learning help improve the performance of the model and make it more robust by reducing the prediction errors over time. This might sound very familiar to the process of training a model on a given dataset, and it also extends to fine-tuning a model on new data, with potentially different distributions. One of the biggest pitfalls in letting the model provide the feedback is declining quality over time. If the model is not perfectly accurate (which rarely happens), using the feedback it outputs on new data might lead to wrong labels. Using this labels to train (in an active learning scenario), will lead to very low accuracy scores after a few iterations.

The second type of feedback loop, as emphasized by [17] (the helpful feedback loop), implies bringing an external entity to help assess the model performance on new data. This can mean a different model, trained to assess the quality of the first, users giving feedback on the predictions they got (for example whether a book recommendation was accurate or whether an ad was relevant) and even data analysts looking at the results and taking decisions based on what they discover. There are many examples of companies gathering user feedback at scale, especially in businesses with high revenues, like the ads industry [10].

Feedback is useful in many ways in large scale machine learning systems. In the research questions of this thesis, we are especially interested in how engineers that manage such systems can adapt them based on data collection feedback. For example, they can get informed that augmenting one specific class of data might bring X% in accuracy increase. Similarly, they might find out that gathering data for another class won't increase the quality of the model. In this way, we aim to boost the productivity of such people, by using their time towards the action that is likely to bring the largest benefit, instead of spending time on low/no quality tasks.

The literature on this topic is quite shallow and follows standard procedures in retraining machine learning modes. Without thinking of ways to optimize the flow, engineers either (re)train the model on new data, when, there might be a way to substantially speed the flow by pruning it. Following, we review some of the relevant papers on this matter.

Given that NLP is quite an active topic at the moment with advancements in ChatGPT and similar language models, we decided to start the review on data collection feedback with a paper on NLU (natural language understanding). In [31], Parrish et al. look into ways to enhance the quality of the data they collect (and of the subsequent model) by adding linguists in the loop. Even if the concept of a linguist is highly specific to this domain, the technique of adding an 'expert' on the subject is practiced in other areas as well. The goal in the paper is to raise the quality of crowdsourced NLP datasets, reduce the gaps in data and correct any potential biases, by augmenting the work done by non-expert annotators, and having experts guide them in ways that would address issues in data.

The reason why they focus on the language task is the inherent subjectivity that

comes with the domain; multiple non-expert annotators can have different views over a sentence, with different thresholds when it comes to deciding whether it is a contradiction, for example. They experiment with 3 scenarios: one with no expert involvement (data coming directly from the 3rd party source), one with light expert involvement (the 3rd party source follows some predefined guidelines), and one with more expert involvement (monitoring and close discussions between the expert and the 3rd party provider).

They conclude that, for this specific task, using an outside entity to monitor and course-correct the process helps with the quality of data, which has a direct influence on higher accuracies of the model it is fed into. An open question that they frame as future work is how to take the same approach, but with the aim of not only improving the model accuracy on the dataset at hand but also exploring methodologies to enhance generalizability.

Domain experts are invaluable to any data collection process for machine learning, as their expertise in the domain can identify nuances that non-experts may overlook. This can help to identify biases in the data and reduce gaps in the data that could lead to incorrect or incomplete models. The guidance domain experts provide to non-experts can help to ensure that the task is completed correctly and to the desired standard, resulting in data of the highest quality. In addition, the presence of domain experts in the data collection loop can improve the accuracy and performance of machine learning models, as the quality of the collected data will be improved. This provides an additional benefit to any organization that is seeking to leverage machine learning technology, as the quality of the data collected will directly impact the success of the project.

In this thesis, we are not aiming to introduce a human supervisor in the process, since that will slow the flow even more. Instead, we plan to create automated tools for giving feedback on data collection, in relation to the model it is trained on. In doing so, we aim to achieve smaller datasets, that perform as well as larger ones on models.

3.2. The Enterprise Machine Learning process

For setting the higher level scene on what this thesis aims to accomplish, it is wise to look at the bigger picture of applying machine learning at scale, and understanding how important the data component is and how this research can help improve the overall process.

Machine learning is becoming increasingly popular in the enterprise sector as it allows companies to gain valuable insights from large amounts of data. This can be used to optimize business processes, improve customer service, and drive revenue growth. One common application of machine learning in the enterprise is predictive modeling, which uses historical data to make predictions about future events. This can be used to forecast demand for products, identify potential fraud, or detect patterns in customer behavior. Additionally, machine learning can be used for natural language processing tasks such as sentiment analysis, which can be used to analyze

customer feedback and improve customer engagement.

The machine learning process at a big scale follows more or less the same sequence of steps:

1. **Setting the expectation and planning the project:** This includes deciding what is the end goal, together with the metrics for success and potential alternatives in case the result is not as expected.
2. **Gathering data or aggregating existing data from various sources:** The company can either have data they want to train a machine learning model on, or they plan to gather it from other sources. In the first case, the data is possibly in different formats and stored in various containers (like data lakes, databases, and unstructured buckets of data). In the second case, they can start the process of data gathering, either using their own systems or fetching it from a 3rd party vendor.
3. **Cleaning the data, removing redundancies and making sure it is relevant for the task:** This is the part this thesis focuses on, right in the step where data is available, and right before it is being fed to a machine learning algorithm. Many steps of data quality and data selection process will be handled extensively in the thesis, but at a high level, the goal is to minimize the data quantity by removing redundancies, enhancing the data by offering insights on how it should be better collected, and selecting the most relevant samples with the goal of making the next steps more efficient. All this, while maximizing the quality of the model.
4. **Pick a model:** This step highly depends on the usecase and on the type of data collected. Popular models include neural networks, SVMs and decision trees, but the way they are used is highly dependent on the task at hand. With the rise of Automated Machine Learning [6], this step can be largely automated nowadays, saving lots of time that could be otherwise spent in previous data quality steps.
5. **Train the model and deploy to production:** Once the model is trained on high quality data, this is the step where it is actually included in the product, usually as a prediction API, that can be used to serve product features like detecting fraud, doing better product recommendations to customers and making real time predictions for autonomous vehicles, for example.
6. **Monitoring and retraining on new data:** As the product changes and the data distribution shifts, the model needs to be monitored and periodically trained against new, more up to date data. This can be seen as a reiteration of steps 3 to 5, with same goals, but an updated dataset.

As a recent study [21] shows, while machine learning can be powerful in many situations, there are a couple pitfalls that many big companies fall into. A few examples include picking the wrong usecase, the wrong data and the bias associated with the dataset they are using. We presume that the quality of data, and picking a representative sample of it, can make the difference in large scale machine learning.

A recent article on the MLOps process [37], which discusses the process of putting machine learning models into production, emphasizes the need to look at a problem from a system view, not a model view. In the research world specifically, machine learning scientists/engineers develop a model in isolation, usually within a Jupyter notebook or a similar environment that allows for quick experimentation. According to a study from Sculley et al. [37], the model development part is just a small and isolated component into a larger infrastructure. When viewed from a system level, there are way more variables to think about, from data ingestion, to data quality, monitoring, scalability and alerting, that makes the ML component relatively small. This does not mean that it is not important. Developing high quality machine learning models is crucial for making a AI backed systems work, but the model development part has to be done having in mind the system as a whole, with the challenges or real world data and having the product used by actual users, that might act in different ways given different predictions of the machine learning model.

A study from Forbes [13] shows that the work of data scientists is mostly focused on data quality related tasks, and not on training machine learning models. This can easily be translated in the implication that high quality data will most likely drive the development of good machine learning models, and making sure the data is relevant is a crucial step in the process. According to [13], “Data preparation accounts for about 80% of the work of data scientists”, and “57% of data scientists regard cleaning and organizing data as the least enjoyable part of their work”.

This brings more motivation to this thesis, with the goal being to find means to automate data cleaning and selection, to a point that the data is relevant and ready to be passed to a machine learning model.

4
Systematic Literature Review

As a base for the practical work to be done in later chapters, we did a systematic literature review (referred to as SLR), to dive deep into the topic at hand and explore the literature on it. We aim to understand common patterns, challenges, and areas of expansion. Starting from an initial pool of 377 papers, we applied specific inclusion and exclusion criteria and did 2 iterations of snowballing. The goal was to find papers that aim to reduce the dataset, either through feature selection or data point sampling, and contribute to both reducing the time of machine learning iterations, as well as improving the quality and interpretability of the model. Given this goal, we extracted and analyzed a total of 36 papers, ranging over 8 domains, outlining 3 main data selection approaches, multiple ways of reducing the size of the dataset, and various approaches for picking better data sources.

4.1. SLR overview

The SLR conducted in this thesis aims to provide a comprehensive and structured overview of the existing research on data selection, feature selection, and time reduction in big machine learning workflows. The SLR methodology allows for the identification, evaluation, and synthesis of relevant literature in a systematic and reproducible manner, ensuring a transparent and documented review process. By analyzing the current state of research, this SLR identifies key methodologies, techniques, and challenges, as well as potential avenues for future research. It offers hints on what have been the trends of the papers on the topic in the last decade, as well as what domains benefit the most from a reduced and more insightful dataset.

4.2. Methodology

4.2.1. Search query and search engine

A comprehensive search strategy was developed to identify relevant research articles from multiple electronic databases, including IEEE Xplore, ACM Digital Library, Google Scholar, Scopus as Web of Science. The search process was guided by a set of predetermined keywords and phrases, aiming to highlight recent papers that are relevant to the machine learning field and are related to data selection, clustering, or pruning, as those topics align with the main research questions of the thesis.

We did a couple of iterations on the search query with the goal of identifying specific papers, but also keeping the search generic enough to cover relevant data selection studies from multiple industries.

Table 4.1: Paper query iterations

Iteration	Query
Initial query	(”machine learning” OR ”ml” OR ”mlops” OR ”data engineering” OR ”machine learning pipeline*” OR ”ml pipeline*” OR ”data centric”) AND (”data* pruning” OR ”data* selection” OR ”feature* selection” OR ”feature* pruning) AND (”collection feedback” OR ”data feedback”) AND after: 2015
1st iteration	(”machine learning” OR ”mlops”) AND (”data” OR ”dataset” OR ”feature”) AND (”selection” OR ”clustering” OR ”pruning”) AND after:2015
2nd iteration	(”machine learning”) AND (”data clustering” OR ”data selection” OR ”dataset pruning” OR ”feature selection”) AND (”selection” OR ”clustering” OR ”pruning”) AND after:2015
Final query	(”machine learning” OR ”mlops”) AND ”data” AND (”selection” OR ”clustering” OR ”pruning”) AND after:2015

Because of the large number of variations inside the clauses in the first query of Table 4.1, the number of papers was often larger than a couple hundred, which was an unmanageable amount, as well as the papers would have been too generic. After a couple of iterations, with the goal of shortening the query, while still capturing relevant papers, we ended up with the final query in the same table. We decided to focus on papers after 2015 in the first iteration, as the last years have brought an increase in both data and awareness of the importance of having a quick and relevant machine learning pipeline. However, in both iterations of snowballing, we picked papers regardless of year.

We used the Publish or Perish tool to search the titles related to our research topic, and we ended up with a 377 paper list. While we acknowledge that this set might not be entirely comprehensive or include all the interesting literature studies, it is still a significant number that we can work with. In addition to this, we employed a 2-step snowballing technique to capture any papers that may have been missed in our initial query. This technique allowed us to expand our search beyond the initial set of papers and discover additional relevant research.

4.2.2. Inclusion/exclusion criteria

As the search query is generic, a large number of irrelevant papers were gathered in the process. The following are the inclusion criteria:

- **I1**: Shows potential to reduce the time of an ML model/pipeline.
- **I2**: Talks about reducing the dataset.
- **I3**: Offers a mechanism that provides feedback on data collection.

And the following are the exclusion criteria:

- **E1**: Not in English.

- **E2**: Full PDF not available.
- **E3**: Talks about data, but not in a machine learning context (i.e. only about data structures, only about databases, etc).
- **E4**: Not an academic paper.
- **E5**: Not peer-reviewed.
- **E6**: A duplicate.

In order to qualify for the selection, each paper had to show potential to reduce the time of an ML model/pipeline (**I1**), either talk about reducing the dataset (**I2**) or offer a mechanism that provides feedback on data collection (**I3**) and have no exclusion criteria checked. This leads to the following formula:

I1 *AND (****I2*** *OR* ***I3****) AND NOT (****E1*** *OR* ***E2*** *OR ...* ***E6****)*

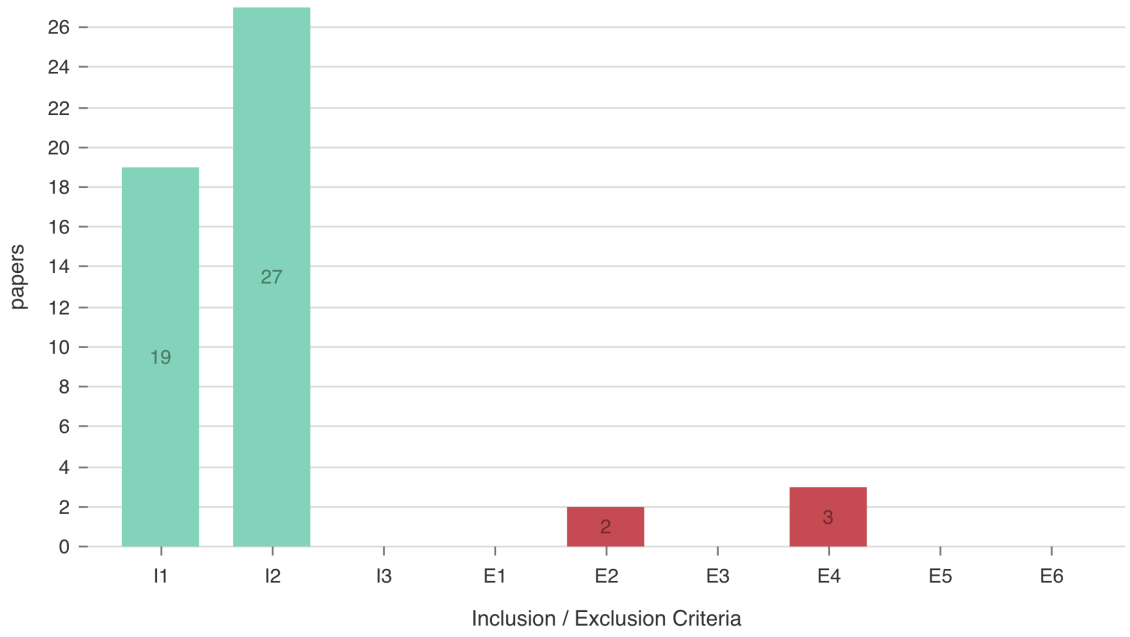

Figure 4.1: Paper count by Inclusion/Exclusion criteria

Out of the initial 377 papers, only 19 were included. The majority of the exclusions were either irrelevant, didn't have the PDF available, or were not academic papers (blog posts, for example). There was no paper to qualify for the 3rd inclusion criteria, which is not a surprise since the search query does not directly address the notion of providing a mechanism for data collection feedback. The most generic inclusion criteria, I2, was most often checked off, while there were only 19 papers that showed the potential of reducing the time of machine learning flow iterations (I1).

It is interesting to note that, as represented in Figure 4.1, all the papers that comply with I1 also comply with I2. Based on our observation, when the selected papers discuss reducing the time of machine learning processes, they also talk about reducing the dataset size.

4.2.3. Snowballing

The snowballing process involved examining the reference lists of the papers we found in the initial search, as well as looking at the papers that cited them. This approach helped us identify a large number of additional papers that were not initially captured in our search, ultimately resulting in a more comprehensive and robust set of literature to draw from for our research. We performed both forward snowballing (examining papers that cite the paper being examined), as well as backward snowballing (examining papers that are cited by the paper being examined).

Each snowballing iteration was done on a curated list of papers, after applying the inclusion and exclusion criteria. First, we had the initial set of 377 papers, that we applied the inclusion and exclusion criteria on. This led to 17 papers, that generated another 12 after the first iteration of snowballing and applying the criteria. Finally, the second snowballing iteration generated the rest of 7 papers.

4.3. Paper Attributes and Questions

4.3.1. Attributes

For each paper, we tracked a predefined list of attributes, that would help us later to gather papers into specific buckets, analyze each group, and draw aggregate conclusions across multiple papers that are similar. The following attributes have been tracked:

- Research type

  Refers to the type of study, based on the title and content. Can have one of the following values:

Table 4.2: Research types

Solution	The paper discusses a novel approach, explains the algorithm, and provides proofs and analysis on the new solution provided. Those papers usually refer to studies and previous solutions, and either builds a completely new approach, or improve existing ones.
Review	The paper presents multiple data selection/ML flow time reduction approaches, and compares them at a theoretical level. Reviews tend to not go in depth into results or algorithms, but instead offer a high level view of different approaches.
Analysis	The paper presents multiple approaches, with the idea of having an objective comparison, that involves running the algorithms and presenting the results side by side. They usually tend to use benchmark datasets, and present what algorithms perform best in a setup where multiple hyper parameter configurations are used.
Other	The paper talks about data selection or about reducing the time of machine learning flows, but does not bring a new solution nor present existing ones.

- Domain

  Refers to the business domain of the paper. If it's a solution, it refers to the specific domain in which it is applied. If it's a survey, it typically refers to the type of datasets that various algorithms are tested on. Each paper belongs to one of the following:

Table 4.3: Domains

Agnostic	The paper presents a solution that is not specifically linked to a domain, it's more versatile and can be applied on a wide range of tasks.
Manufacturing	The paper talks about data reduction in the goods manufacturing industry.
Movie review	The paper presents feature elimination strategies for learning accurate machine learning models that rank/review movies.
Construction	The paper presents approaches for machine learning time reduction and feature elimination in the construction industry.
Computer vision	The paper deals with algorithms that run on visual data (images, videos), and aims to reduce the feature set to speed up learning.
Health	The paper presents feature elimination strategies in the health industry, most common in DNA sequencing.
Networking	The paper talks about identifying the most important features in the networking domains, such as identifying malware, phishing attempts, and network intrusion detection.
Financial	The paper talks about reducing financial datasets.

- Tags

  Papers can be split in multiple categories, that don't have to be necessarily disjoint. This helps in creating discussion points that employ only a specific subset of the papers. The tags assigned to papers, split into categories, are:

  - Tags that refer to how the data is selected (Table 4.4)

Table 4.4: Tags for Data Selection

Feature Selection	The paper reduces the dataset by picking a subset of the existing features in the dataset.
Data Source Selection	The paper aims to improve the dataset by picking better data sources.
Data Point Selection	The paper aims to reduce the length of the dataset by picking only the rows/samples that provide the most input.

- Tags that refer to the category of the paper (Table 4.5)

Table 4.5: Tags for Categories

Overview	The paper is an overview of multiple approaches.
SLR	The paper offers a systematic literature review.

- Tags that refer to the specific ML algorithm used (Table 4.6)

Table 4.6: Tags for ML algorithm

Regression	The model involved predicts a continuous value.
Classification	The model classifies the datapoints as belonging to one class.
Clustering	The model splits the datapoints in an unsupervised way, into different clusters.

- Tags that refer to the industry type (Table 4.7)

Table 4.7: Tags for Industries

Industry specific	The paper discusses a solution that can be used only in a specific industry or setup.
Industry agnostic	The paper offers a generic algorithm or overview.

- Tags refer to the specific use cases that span across multiple papers (Table 4.8)

Table 4.8: Tags for Specific Usecases

Intrusion Detection	The paper deals with networking data and aims to detect anomalies such as intrusion detection in networks.
Credit Scoring	The paper deals with financial data, aiming to create models that predict the credit score.

- Feature Elimination approach

  One of the most interesting topics for this SLR is the feature elimination (FE) strategy, hence the tag. As also discussed before and as we'll dive deeper into in later chapters, there are 3 main FE approaches:
- Time reduction strategy

  Another important topic for the SLR is how, each of the 19 papers that talk about reducing the time in machine learning flows, aim to reduce the size of the dataset. The initial categories were the following, even if after a first pass, we could not find papers for some of them.

  Given the specific search query, it is of no surprise that most of the papers belong to the Data centric category.

Table 4.9: Feature Elimination Approach

Filter	A filter approach aims to pick a subset of the features based on some standalone scores, unrelated to the model the data will be fed into.
Wrapper	As opposed to filter, a wrapper method decides what features should end up in the final dataset based on information from the model itself.
Embedded	Embedded feature elimination tools imply that the model does feature elimination itself. One example is the dropout method in neural networks.

Table 4.10: Time Reduction Strategy

Data centric	The paper aims to simplify the dataset as a way to speed up ML flows.
Model centric	The paper fine-tunes the model to speed up ML flows.
Business centric	The paper looks into business decisions that might influence how fast ML iterations are.
None	The paper does not attempt to reduce the time of ML flows.

- Has code

  We were also interested in whether papers have a replication package that we can try within this SLR.

4.3.2. Questions

We aimed to answer 3 questions on each paper so we get insightful information for the Research Questions of the thesis. The questions are:

1. How does the paper reduce the overall time of ML pipeline iterations?

   Time reduction can occur in many ways, from randomly reducing the dataset, to model simplification or either early stopping of the flow. Our goal is to zoom in the data part, as we are specifically interested in how the dataset can be filtered (vertically or horizontally), to support faster machine learning iterations, keeping the quality of the resulting model high.
2. How does the paper attempt to reduce the dataset?

   As outlined before, there are multiple ways to reduce a dataset, from very trivial such as random selection, to more advanced, such as filtering based on information gain of each sample or feature. This question provides insights on what is the approach of the paper to reduce the dataset that enters the machine learning flow.
3. Does the paper use/prefer any of the filter, wrapper or embedded FE methods?

   While reading the papers in the SLR, 3 main feature selection categories prevail, and we decided to include this question for all the papers, to get a better sense of

why papers choose one category over the other. They all come with pros and cons, as will be outlined later.

4.3.3. Notes format for each paper

Other than the questions above, we filled in additional information for each paper. We noted down the key points, ideas, and results, as well as notes about the paper quality itself, such as how clear it is, how can the information in the paper improve this SLR, and whether the paper falls into a specific category.

4.4. Final Paper List

Given the initial paper list that came as a result of the search query on Google scholar, applying the selection criteria, doing 2 rounds of backward and forward snowballing, and applying the selection criteria on the snowballed papers, we ended up with a list of 36 papers. All of them show the potential of reducing the time of Machine Learning pipelines, as well as reducing the overall size of the dataset. None offers a mechanism that provides feedback on data collection. The study and discussions below are based on this set of papers.

Below is a visualization of the distribution of papers into types of research, industry, and the correlation between them. Our discussions below will be based on such category splits, discussing insightful groupings and summarizing the findings from each. For example, it's clear from the graph (Figure 4.2) that many of the papers provide Agnostic Solutions.

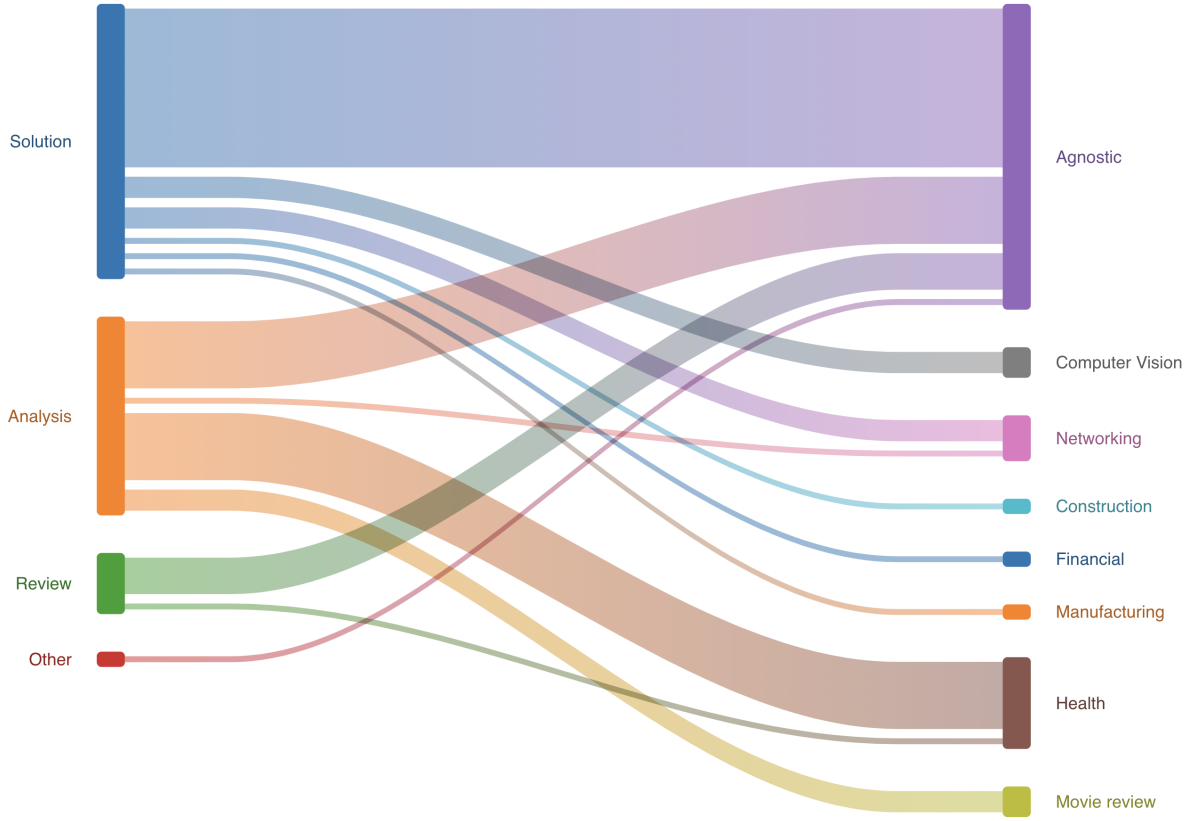

Figure 4.2: Paper count by research types (left) and domains (right)

4.5. Preliminary Insights

In this section, we summarize the key findings from our initial review of the literature. It covers the main research types and domains identified, along with the tagging approach used for categorizing the papers. This overview establishes an early understanding of the landscape of feature selection and its role in optimizing machine learning processes.

1. Research types

   After the initial pass through the selected papers in the systematic literature review, several key insights and common themes have emerged regarding feature selection and the acceleration of machine learning workflows from a data perspective. This section presents a summary of these observations, providing a foundation for further analysis and discussion.

   Feature selection plays a pivotal role in streamlining machine learning workflows, improving model performance and revealing insights from the data. It has become increasingly essential as the dimensionality of datasets grows in the era of big data. The primary benefits of effective feature selection include:

   1 Reduced computational complexity: By selecting a subset of relevant features, the dimensionality of the dataset is reduced, resulting in lower computational demands for model training and inference. Sometimes, data acquisition can be expensive as well, from both a time, as well as a financial perspective. For example, [42] outlines the importance of picking the right dataset from the beginning, as not only the cost of picking can get high, but the cost of changing it later is even higher.
   2 Improved model performance: Removing irrelevant or redundant features can lead to better generalization, reducing the risk of overfitting and enhancing the model's predictive performance. For example, [36] conducts a study that analyses the impact of iteratively pruning the dataset on the performance of a model.
   3 Enhanced interpretability: A reduced feature set simplifies the model, making it easier to understand and explain the relationships between input variables and the target variable. For example, [48] mentions how the real world data is inherently noisy, with many irrelevant and misleading features. Removing them helps not only the performance of the model but also the interpretability.

2. Domains

   Incorporating domain-specific knowledge in the data selection process can enhance the effectiveness of the chosen techniques. Domain experts can provide valuable insights into feature relevance, data point importance, and data source reliability, helping to guide the selection process and improve the overall efficiency and performance of the machine learning workflow. Collaborating with domain experts and leveraging their expertise can greatly benefit the development and application of data selection techniques. However, being domain specific comes at a cost. One one hand, you are developing and algorithm that is overfitted to one domain and on the other, domain knowledge can be expensive to acquire. However, many companies prefer specialization over generalization, since they want to provide the best services into one specific niche.

   In the papers from this SLR, many of the studies have a domain agnostic approach. However, some of them do specialize in a particular domain, such as health, finance and networking. For example, some papers talk about network intrusion detection [35, 8, 7] some talk about microarray DNA data in a health context [26, 15], and one talks about finance [48].

3. Tags

   Tags have been assigned in a semi structured way, with each paper being assigned multiple tags, based on the content and the approach.

   For example, an interesting discussion can be developed across the data selection tags, namely “Feature Selection”, “Data Source Selection” and “Data Point Selection”. Similarly, an insightful comparison will be conducted across the type of model used, using the papers that have the following tags: “Classification”, “Regression” and “Clustering”.

   Also, the co-occurrence of tags reveals patterns about the common topics in the surveyed papers. We will look into tags that occur together, 2 of them being “Feature Selection” and “Industry Agnostic”, co-occurring on 13 papers, signaling that when it comes to removing features, most authors would likely propose a generic approach, rather than one that is tied to a specific industry.

4.6. Research types

This section is structured to offer an understanding of various research types identified in the papers, including Solutions, Reviews, Analyses, and Other approaches. Each category reveals distinct insights into feature and data selection methods within machine learning, shedding light on their applications, benefits, and limitations across various domains. From innovative solutions enhancing machine learning workflows to in-depth reviews and analytical studies of specific methods, this section provides a detailed exploration of the current state of research in this field.

1. Solutions

   The papers in the ”Solution” category primarily focus on novel techniques and methodologies for data selection, specifically emphasizing the importance of selecting the right dataset, feature selection, and reducing dataset size. A variety of approaches and applications are discussed across these papers, including Monte Carlo-generated data for simulating construction stability [44], feature selection in computer vision [53], dataset distillation [22], and feature correlation and stability-based methods [38].

   Many of the papers present new methods for feature selection, aimed at maximizing the relevance to the predicted column while minimizing redundancy between variables [49]. Some of these approaches involve clustering, minimum spanning trees [40], rough sets, tabu search [48], particle swarm optimization (PSO), and evolutionary flows that combine both wrapper and filter techniques.

   A few papers delve into specific applications, such as feature selection in steganalysis using the Mahalanobis distance and support vector machines (SVMs) [15]. Others provide mathematical proofs for wrapper feature selection methods, highlighting their theoretical foundations [23].

   Selective sampling and feature interaction are also discussed in the context of measuring feature relevance and analyzing multiple datasets [29], [55]. Additionally, one paper explores a non-deterministic polynomial (NP) approach to finding the best feature set using backtracking on small sample sizes [51].

The Solutions surveyed in the SLR generally agree that feature collection can be expensive, and doing the selection early in the process comes at a significantly lower cost. When it comes to picking the right data points to learn from, most of the papers focus on “hard-negative mining” [22], as expected, since the hard-to-learn samples are the ones that a model should focus mostly on. However, it is also agreed that hyperparameters of both selection heuristics and the models they are fed into can be a huge cause of instability of the result [44]. Also, removing the noise is a general priority.

Also, given the 3 Feature Elimination methods (Embedded, Filter, and Wrapper), there are just 8 papers that use 2 in their solution and 3 that mention all of them. Below is a co-occurrence graph for those methods on the Solutions in the SLR, showing that Filter & Wrapper (9 papers) go better together than using Embedded & Wrapper or Filter & Embedded (4 papers). The reason why Filter is more popular, as well as the co-occurrence with the Wrapper method might be due to computational complexities. Based on what the papers present, the Filter method is the fastest, followed by the Wrapper and the Embedded method. The Embedded method is the least used one, which might indicate either a gap in research or the involvement of high computational cost.

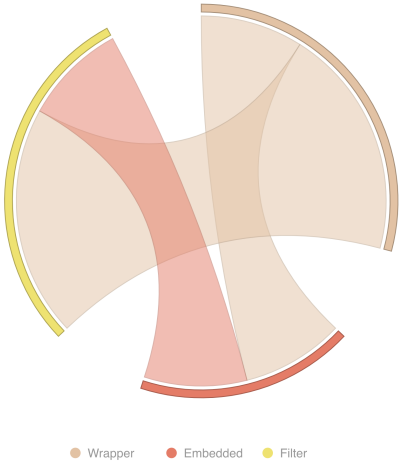

Figure 4.3: Co-occurrence of FE approaches

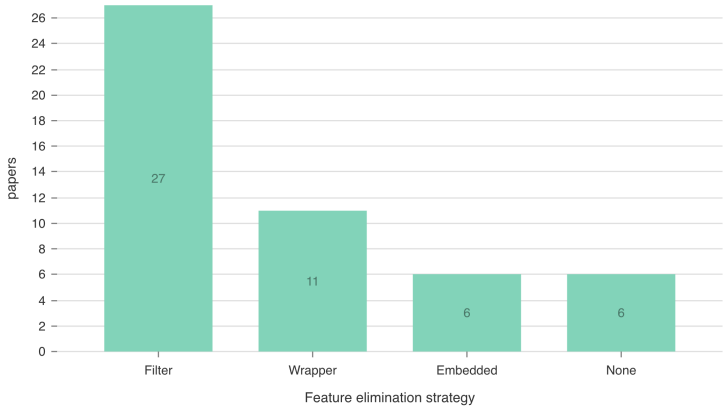

Figure 4.4: Count by Feature elimination strategy

One common pattern in this category is the use of heuristics to determine an optimal feature set. This is typically achieved through a correlation metric or a statistical measure that ranks the features. Although one paper [51] attempts to solve the NP problem of feature selection by backtracking on smaller batches of features, it still acknowledges that feature selection can easily become an NP-hard problem.

The computational cost of finding the globally optimum solution may be the reason why many papers settle for something close to perfect, acknowledging that reaching perfection comes at a too high cost. For instance, papers often state that the filter method is preferred due to its computational efficiency when compared to the wrapper or embedded methods [15, 29].

It is surprising that few papers discuss the feature selection problem in the context of an evolutionary flow or a population-based method. As [54] emphasizes, ”PSO

was not applied extensively in feature selection". This may represent a gap in research that could be explored in the future. With computational power growing and simulations becoming easier to perform, this could lead to interesting future work.

With that said, the solutions presented in this systematic literature review (SLR) aid in answering the research questions of the thesis by highlighting opportunities for future research. However, the paper set of this SLR has gaps when it comes to combining techniques or attempting solutions that converge to better global optima.

In summary, the "Solution" category encompasses a diverse range of papers that contribute innovative techniques and methodologies for data selection in various contexts, emphasizing the importance of selecting the right dataset and employing effective feature selection methods to improve machine learning workflows.

2. Reviews

The papers in the "Review" category primarily focus on feature selection methods and their applications in various contexts. Their approach is to mainly survey existing solutions and provide an overview of them, without getting in-depth to measure the performance of the respective algorithms. Most of the reviews are industry agnostic, while one of them is concerned with measuring physical activity data in the health industry.

One study delves into specific applications of feature selection, such as clustering physical activity data and real-time systems [24]. They highlight the crucial role of feature selection in reducing prediction times and improving model robustness. PCA and correlation feature selection emerge as popular choices in these applications.

Another paper discusses broader aspects of feature selection, such as its role as a feature understanding and knowledge discovery tool [25]. They emphasize the need for understanding data to make informed decisions about which features to keep and discard. The stability of feature selection algorithms, which refers to their resilience when new data points are added, is also discussed. They claim that "A feature selection algorithm is stable only when it produces similar features under the training data variation. Ignoring the stability issue of the feature selection algorithm may draw a wrong conclusion".

In the context of big data, no single algorithm is universally applicable for data selection, classification, and clustering. Instead, hybrid algorithms are preferred, and trial and error remains the primary method for adapting to diverse datasets. Data preprocessing and the extraction of relevant information are emphasized as essential steps in the process. Neeraj et al. [30] splits features into "high weight" (most relevant and non-redundant), "medium weight" (somehow relevant but non-redundant), "less-weight" (redundant), and "zero-weight" (completely irrelevant or noisy).

Subset selection for regression (SSR) techniques is also discussed by [27], with a focus on creating sparse models, overcoming overfitting, and improving model interpretability. Convex optimizations, greedy algorithms, Lasso, and recursive feature elimination are mentioned as methods for achieving SSR.

Comparing reviews to solutions, they offer a broader view of the industry, covering a wider range of topics and approaches. To reinforce the points made by the previous category, reviews highlight that feature selection is both a performance-increasing method and a knowledge discovery tool for the relevant domain. While examining the papers, there appears to be a research gap in comprehending the potential business impacts of data pruning and how it fits into large-scale products and ML pipelines. This might be linked to the confidentiality issue of making company proprietary information public, or it might indeed signal a need for researching the relation between feature selection and dataset engineering in relation to bigger flows, especially encountered in large corporations, with complex products. However, this does not detract from the significant advantages of using feature selection methods in data analysis.

In conclusion, the ”Review” research type offers valuable insights into various feature selection methods and their applications, highlighting the importance of removing noisy data, ensuring algorithm stability, and understanding the data in order to improve machine learning workflows.

3. Analyses

This chapter discusses the analysis of various feature and data selection methods used in machine learning, with a focus on medical data, neuroimaging datasets, network intrusion detection, and other specific domains. We will explore the importance of these methods in reducing computational time, improving model accuracy, and addressing challenges posed by high-dimensional and imbalanced datasets. It is important to note that analyses, as opposed to reviews, provide a deeper understanding of the quality of each method, including benchmark data and a more detailed analysis of the results and tradeoffs of each method. Additionally, industries are more evenly distributed in this category, with a larger share among medical and networking domains.

According to a couple of analyses, feature selection aims to reduce the number of irrelevant and redundant features, resulting in benefits such as improved data visualization, data understanding, reduced training time, and enhanced model performance, as outlined by Song et al. [39]. Data selection differs from data cleaning in that it focuses on choosing the most representative features to capture the entire distribution, while data cleaning addresses issues like missing values and duplicate rows. Dataset selection is crucial for maintaining diversity and balancing the size and computational demands of the learning process [7].

A significant challenge in feature and data selection is the ”no free lunch” theorem [47], which states that no single method can achieve maximum accuracy across all datasets. Furthermore, the optimal subset of features may not be unique [12], and the optimal set of hyperparameters for feature selection algorithms is difficult to determine in practice.

In medical and neuroimaging datasets, having prior knowledge of the disease morphology can be beneficial [47], but feature selection remains essential due to the high dimensionality and potential for overfitting. Similarly, network intrusion detection faces challenges with highly imbalanced datasets, necessitating the use of oversampling and undersampling techniques [7].

It is important to note that existing research on feature and data selection methods often lacks a deeper understanding of the impact of various techniques on the business they are applied to. While benchmark datasets can provide a good picture, it is fair to say that as far as this thesis is concerned, conducting an analysis on a few large-scale datasets and real-world machine learning flows may prove to be highly advantageous.

Moreover, the choice of performance metrics, such as classification accuracy and training time, may not always reflect the true impact of feature and data selection methods on model performance. Some other metrics such as the impact on end users, ease of use, and real-world relevance, even though less quantifiable, would provide valuable input to research. Finally, the sensitivity of different algorithms to iterative data pruning remains an open question, with some studies reporting minimal impact on performance while others observe significant changes.

It is important to note that feature selection requires iterative steps and multi-part approaches for efficient flows. The process involves removing absolute noise first, then delving deeper into what is increasingly important. It is worth mentioning that few papers [12] have noted a potential gap in the use of ensemble methods. While standalone methods may work well, combining complementary feature selection approaches can yield superior results. As discussed in the previous chapter on reviews, a future direction for research may include population-based methods. Currently, only one paper in the analyses section talks about such methods [34].

In summary, feature and data selection methods play a critical role in improving the efficiency and effectiveness of machine learning models, particularly in complex and high-dimensional problem domains. While there is no one-size-fits-all solution, analyzing the specific challenges and limitations of various methods can guide researchers in choosing appropriate techniques for their datasets and problem domains. Further research is needed to address gaps in the current literature and develop more robust and generalizable approaches for feature and data selection in machine learning.

4. Other

While most of the papers were a good fit for solutions, reviews, and analyses, one paper, mostly focused on big data and large machine learning flows, did not have as many common denominators as the others, thus it was placed in its own category.

[1] discusses large machine learning workflows in the context of big data. While it is a good study regarding the challenges of big data, it does not offer a mechanism for reducing the dataset or time in corporate-level machine learning processes. However, it is included in the study for its discussion on the importance of data-driven optimization and notes on learning from uncertain and incomplete data.

As real-world data can be very unstructured, a potential research gap to bridge is how to perform feature selection on unstructured data and efficiently learn from it to build meaningful machine-learning algorithms that provide value to consumers.

4.7. Time Reduction Strategy

This section discusses the findings from the systematic literature review, specifically focusing on how the selected papers address the acceleration of machine learning workflows through feature selection, data point selection, and data source selection.

1. Feature selection

   Feature selection is a critical and essential technique for machine learning workflows. This is because it effectively reduces the number of input variables while retaining the most relevant information, thus speeding up the workflow process. In the reviewed papers, different feature selection techniques were described, including filtering, wrapper, and embedded methods. Filtering methods aim to eliminate irrelevant or redundant features based on statistical measures or other criteria. Wrapper methods, on the other hand, use a supervised learning algorithm to evaluate subsets of features based on their predictive power, in combination with the model. Embedded methods incorporate feature selection as part of the model-building process, resulting in a more efficient and optimized model.

   1 **Filter methods**: These techniques evaluate the relevance of each feature independently, using statistical measures such as mutual information, information gain and correlation. Examples include the work of [40], that proposes a minimum spanning tree approach, and [24], that attempts to cluster physical activity data.
   2 **Wrapper methods**: In contrast to filter methods, wrapper methods assess the predictive power of feature subsets by employing a specific machine learning algorithm. One noteworthy examples include the use of genetic algorithms, in comparison to PSO and information gain [34]. In addition to what we discussed so far, both the scarcity and potential of population based methods is yet again visible here as well.
   3 **Embedded methods**: These techniques incorporate feature selection as an integral part of the learning algorithm, enabling simultaneous feature selection and model training. It is of no surprise that not many papers mention or use them, due to their computational inefficiency. For example, [25] highlights that the reason why embedded methods are not efficient is because the model is trained with the whole dataset to start with, and this can easily lead to bottlenecks in the process, especially on datasets with many features.

   Discussing feature selection is important, not only from an optimisation point of view, but also to understand what are the patterns and directions of the industry. For example, the filter method is by far the most used with the goal of improving machine learning flows. This opens the discussion on whether the embedded method can be improved, or on whether filter & embedded can be used together, with the goal of analysing both the dataset in a standalone manner, as well as taking into account the interaction it has with the model. Moreover, future research might not necessarily classify new methods as filter, wrapper and embedded. New categories might arise, especially with a quick advancement in generative models, that might even suggest what features to acquire based on the weaknesses of the

model. We might see a new category of “feedback driven” or “generative” feature selection/acquisition methods, that make sense of the domain being tackled and generate the need of new feature by themselves.

2. Datapoint selection

   Data point selection, also called instance or sample selection, is a critical process in machine learning that involves choosing a subset of the dataset that represents the population. This can help reduce computational complexity and training time for the model. There are several techniques available in the literature for data point selection. The business domain of the machine learning flow is an important point to make when choosing a strategy, as only 1 in 5 papers of this category is industry agnostic. Distribution shifts, outliers, compliance concerns and peculiar patterns in data make it more challenging for datapoint selection to act as a one-size-fits-all approach. The following are some of the commonly used data point selection techniques:

   1 **Random sampling**: This method involves selecting a random subset of data points from the original dataset. While [22] provides a comparison of their method with random selection, [44] points out that neural networks trained using randomly distributed data are unstable, especially at the tails of the distribution. As the research in the field is quite advanced, we were not expecting random sampling to be effective standalone, but it’s worth mentioning because it might be able to lay the foundation of population based methods.
   2 **Dataset distillation**: [22] aim to reduce the overall time of ML pipeline iterations by constructing a representative set of data points, called the core set, consisting of high-contribution and informative samples. This is achieved through dataset distillation, which creates a representative sample for each class, and by measuring the learning contribution of each sample. The dataset is then selected based on learning contribution, followed by model training and evaluation. The paper uses a combination of filter and wrapper methods, leaning more towards filter methods, to achieve these goals. The approach is tested using the MNIST and USPS image datasets, with promising results when compared to random selection.
   3 **Under/oversampling**: Under-sampling and over-sampling are techniques used to address class imbalance in datasets, which can negatively impact the performance of machine learning algorithms. Under-sampling involves reducing the number of instances in the majority class to balance the class distribution, often by random selection. However, this technique may discard potentially valuable information. On the other hand, over-sampling involves increasing the number of instances in the minority class to create a balanced distribution. This is typically achieved by duplicating existing instances or generating synthetic ones, such as with the Synthetic Minority Over-sampling Technique (SMOTE). Although over-sampling can help address the class imbalance, it may also introduce noise and increase computational complexity due to the larger dataset size, as Bagui et al. [7] points out.

   When it comes to selecting a relevant subset to learn from, a potential research area that has yet to be explored is the use of datapoint selection in real-world

data which can often be unstructured and highly imbalanced. The challenge lies in being able to take in data of any format and assess its relevance in comparison to a fixed end goal. This will enable multi-modal datapoint selection and represent a significant advancement in the field.

3. Data source selection

   When it comes to selecting the right data source for a machine learning task, there are several factors to consider. One of the most important is data quality, which ensures that the data is accurate, complete, and consistent. Another key factor is reliability, which is what separates trustworthy sources from unreliable ones. Finally, relevance is also an essential element to bear in mind, as it allows us to focus on the data that is most pertinent to our task.

   Although the pool of papers in this category is small, they all mention the importance of domain specific knowledge. When it comes to data acquisition, there are multiple aspects to consider, and not only the relevance of the data. For example, a good first question is “Relevant to what?”, followed by the cost of acquiring the data [42], the time it takes to gather it (an API is fast vs having 1M people fill out a survey), as well as the reliability of the acquisition method.

   While more on the business side, data source selection is still relevant for the thesis, as in big companies, the data streams are numerous, and focusing on the right ones can yield superior results. An open research area can be represented by the multi-modality of the data sources, and the goal of creating a unified format, such that a learning algorithm can find meaningful patterns. Figuring out not only how to handle multi-modal data, but also how to find the relevant subset of features inside is for sure a challenge. The advancements in multi modal generative models and, in general, LLMs, can prove valuable here, as a ‘data selector’ model might not only learn from the features, but also from the business decisions that lead to acquiring the features in the first place, being able, thus, to have a meaningful opinion on what to include in a Machine Learning flow.

4.8. Feature Selection Approaches

Feature selection/elimination is a critical step in machine learning pipelines. Its purpose is to reduce the number of irrelevant and redundant features in a dataset. This process can lead to many benefits, including improved data visualizations, better data understanding, reduced training time, and enhanced model performance.

In this chapter, we provide an overview of the three primary categories of feature selection methods: filter, wrapper, and embedded. We will discuss the strengths and weaknesses of each approach, as well as specific techniques within each category that illustrate their characteristics.

It is already clear from previous sections that there is an imbalance in the number of papers that use each method. Since each method has its own strengths and weaknesses, we will focus on understanding how these methods have evolved over time and what the preference was among papers written in different periods. The graph below shows the tendency of using the three methods in the last 30 years. Of course, this is subject to selection bias, since more papers have been written on the

topic in the last 10 years compared to before. However, the absolute numbers are not necessarily relevant; it is the relative percentages that matter.

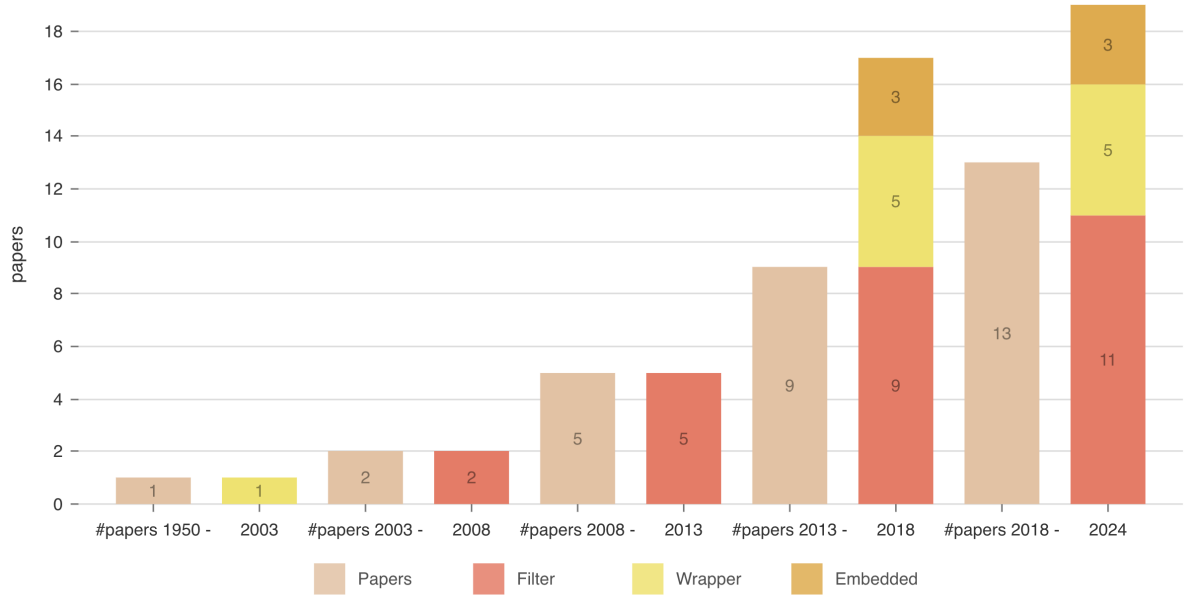

Figure 4.5: Paper count by feature selection approach, in batches of 5 years

Figure 4.5 above shows a clear pattern of growing popularity for Filter methods over the years, particularly as more statistical measures are explored that enable engineers to assess the quality of each feature. The rise in popularity of feature selection, in general, is correlated with the increase in popularity of machine learning and data science as a whole. As more data becomes available, there is more interest in making the most out of it. It can be observed, however, that Embedded methods have only recently been attempted and have already declined in popularity. The graph highlights once again that Filter methods are the most widely used, while Wrapper and Embedded methods follow.

1. **Filter**

   Filter methods are a class of feature selection techniques that rely on ranking features based on their individual importance, usually using a statistical measure. This approach is independent of the learning algorithm, making it computationally efficient and straightforward to implement. However, filter methods may overlook feature interactions, as they evaluate each feature independently.

   These methods are particularly appealing due to their computational efficiency and independence from classifiers. However, some potential gaps and future research directions specifically related to filter methods can be identified:

   1 **Feature Interaction**: It is evident that certain features may possess little individual correlation with the target concept, but when combined with other features, they exhibit strong correlations. Investigating and addressing feature interactions in filter methods could potentially improve their performance. Developing novel algorithms that focus on feature interaction, such as the INTERACT algorithm, may be a promising direction [55].

2 **Handling Imbalanced Datasets**: Filter methods may struggle with imbalanced datasets, where rare events such as network attacks are underrepresented. Future research could focus on improving the performance of filter methods for imbalanced datasets, ensuring that they can accurately learn from the available data [8].
3 **Redundancy Detection**: Some filter methods are effective at eliminating irrelevant features but may not be as successful at detecting redundant features. Developing algorithms that effectively handle both irrelevant and redundant features could improve the quality of feature selection and enhance overall model performance [30, 8, 48], as well as reduce the quantity of the data.
4 **High-Dimensional Data**: As datasets grow in size and complexity, the scalability of filter methods becomes increasingly important. Investigating approaches suitable for high-dimensional data, such as the ranker method, may provide valuable insights into handling large datasets efficiently [40, 5].
5 **Stability and Cost Considerations**: Developing filter methods that consider feature stability, relevance, and collection costs could help optimize the feature selection process. The RABFS (redundancy analysis-based feature selection) [49] and the ScC (hybrid) [38] model are examples of methods that aim to address these issues.

Although filter methods have proven effective in various domains, there are still gaps and areas that could benefit from further research. By addressing these challenges, researchers can develop improved filter methods that better handle diverse datasets and achieve optimal feature selection.

2. **Wrapper**

Although wrapper methods may be less popular and efficient than filter methods, they have one large selling point: they leverage both the dataset and the model. In certain scenarios, a dataset may be so domain-specific that only specific models can find the correlations in the data.

Wrapper methods assess the performance of each feature, taking into account the quality of the model trained on a specific feature set. In comparison to filter methods, they are more powerful as they can identify what features interact and which are redundant. By creating multiple overlapping feature sets and measuring the quality of the model on each, one can determine the intersection of features and optimize the feature selection process.

However, wrapper methods tend to be more computationally expensive than filter methods, as they involve training the learning algorithm multiple times. Future research could explore ways to optimize wrapper methods for large datasets and high-dimensional problems without sacrificing performance. The following are a few future research directions, on notions that concern Wrapper methods:

1 **Scalability and Computational Efficiency**: Wrapper methods tend to be more computationally expensive than filter methods, as they involve training the learning algorithm multiple times. Future research could explore ways to optimize wrapper methods for large datasets and high-dimensional problems without sacrificing performance. For example, using proxy models might be

a good trade-off between the speed but lack of context of Filter methods and the inefficiency but increase of performance of Wrapper methods. Having a lighter model that behaves similar to the main one might reduce the time of finding optimal features, while still retaining some of the context on feature interaction that the model provides.

2 **Stability and Robustness**: Wrapper methods may be sensitive to small changes in the training data, leading to inconsistent feature selections. Moreover, they are directly influenced by the biases and instabilities of the model used. Developing techniques that ensure the stability and robustness of wrapper methods could improve the overall effectiveness of feature selection, such as the regularization feature for preventing neural networks to overfit.

3 **Algorithm-specific Wrapper Methods**: Wrapper methods often use a specific learning algorithm for feature selection, such as neural networks, SVMs, or random forests. Investigating algorithm-specific wrapper methods could lead to better integration and improved performance of the learning algorithm and the feature selection process.

4 **Greedy Algorithms**: The papers discuss the performance of greedy algorithms, such as forward selection and backward elimination, in the context of wrapper methods. Future research could explore ways to optimize these greedy algorithms or develop novel approaches to further improve the effectiveness of wrapper-based feature selection using heuristics.

3. **Embedded**

The papers shed light on the significance of embedded methods in feature selection, particularly in the context of model stability, neuroimaging datasets, and computer vision tasks. Embedded methods offer the advantage of integrating feature selection directly into the learning algorithm, allowing for the discovery of relevant features during the model training process. However, there are still some potential gaps and future research directions related to embedded methods:

1 **Interplay between Feature Selection and Regularization**: Regularization techniques are often used in embedded methods to control the importance and influence of different features. Further investigation could explore the interplay between feature selection and regularization, and how different regularization strategies affect the feature selection process and overall model performance.

2 **Unsupervised Feature Selection**: The papers mention unsupervised feature selection, particularly in the context of clustering. Future research could delve deeper into unsupervised methods for feature selection, investigating their effectiveness in revealing hidden data structures and enhancing clustering performance.

While a less popular approach, with not so many papers mentioning or using embedded methods, they can play a big part in machine learning pipelines, especially if they are optimized to work well for specific scenarios, such as the dropout method in neural networks.

4. **Mixes of methods**

   Integrating multiple methods might yield better performances, without compromising much on performance. For example, [49] uses both filter and wrapper methods, in the context of Binary Particle Swarm optimization. They aim to leverage the benefits of using filter methods by reducing the size of the datasets first and then taking into account the output of the model to further optimize the data selection mechanism.

4.9. Datapoint Selection Approaches

As opposed to feature selection, where the dataset is reduced in width, datapoint selection aims to reduce the length of the dataset, while keeping the feature set unchanged. In large scale machine learning, the quantity of data can become huge, while the relevance might not be a metric that scales with size. The goal is to only keep informative samples in the learning pool, reduce the number of redundant features and preserve hard to learn samples.

1. **Dataset distillation**

   Jeong et al. [22] addresses the issue of datapoint selection in image datasets, specifically MNIST and USPS. The objective is to reduce the size of the training data without compromising the quality of the model. Each dataset contains features that significantly contribute to the predictor's accuracy, as well as features that are less important, following the 80/20 rule. The former category is referred to as ”hard negatives” — samples that represent the boundaries of one class in a classification scenario. They're critical in shaping the model's data splitting ability. [22] terms them the ”core set” in their paper, and aims to include only high-contribution and informative samples, as using a large set only slows down the training process. They suggest that ”uncertainty sampling” is an effective query strategy that selects the samples with the greatest degree of uncertainty, allowing for more information to be leveraged during training. We can extend this by stating that dataset distillation is a ”wrapper” datapoint selection method, as it relies on the model's predicted uncertainty.

   It is important to note from their paper that, in a learning task, selecting the samples that are hardest to learn first is crucial to maximize the quality of the model while minimizing convergence time. As future research, it would be interesting to explore a split for data point selection into ”Filter”, ”Wrapper”, and ”Embedded” categories, similar to feature selection.

2. **Monte Carlo simulations**

   Toneva et al. [44] highlights again the negative impact of redundant data in the speed of the learning process. Moreover, they add that redundant data can overflow outlier samples that can predict edge cases, and keeping only the important datapoints helps neural networks generalise better.

   The authors' approach involves collecting data through physical simulations, which is an expensive method for data acquisition. To mitigate this, they run Monte Carlo simulations to extract the same number of samples for each bin. While neural

networks trained with random data are unstable, particularly around the tail of the distribution, the authors conclude that the hyperparameters of the model are still a major cause of instability, even with well-balanced data distributions.

3. **Dealing with imbalanced data**

   Bagui et al. [7] deals with the issue of imbalanced class sizes in network intrusion detection. The presence of an intruder in a network is an anomaly, and as a result, accurately training a model using this highly imbalanced data poses a challenge. There are two main techniques to address this issue: either resampling the minority class or downsampling the majority one. The problem with the first approach is how to correctly generate new samples, as well as the increased training time for the model. The second approach faces the issue of potentially removing informative and important samples.

   In their paper, they discuss Synthetic Minority Oversampling Technique (SMOTE), a resampling method designed to address the issue of imbalanced datasets in machine learning. SMOTE works by synthesizing new minority class samples based on the existing minority samples. It randomly selects a minority class instance and identifies its nearest neighbors. It then creates synthetic samples by interpolating between the selected instance and its neighbors. This approach helps to alleviate the class imbalance problem by increasing the representation of the minority class in the dataset. By generating synthetic samples, SMOTE effectively augments the training data and provides more balanced class distributions, which can improve the performance of machine learning models.

In the paper pool of this SLR, there is a notable disparity between the number of papers that address datapoint selection and those that address feature selection. While both topics are important for machine learning workflows, it seems that the latter has received more attention in recent years. This may be due to the fact that understanding features and studying feature interactions is crucial for building effective models with large datasets. In contrast, simply reducing training time by sampling the data vertically may not always lead to optimal results and may overlook important patterns in the data. Therefore, it is important for researchers to carefully consider both datapoint and feature selection when designing their machine learning pipelines, as future research.

4.10. Domains

The field of feature selection and dataset reduction encompasses a wide range of domains, each with its unique challenges, requirements, and applications. In this chapter, we delve into the distribution of research papers across some of the most interesting domains to gain insights into the landscape of feature selection within different fields. We explore domains such as health, cinematography, finance, and domain-agnostic approaches, to understand how feature selection techniques have been applied and adapted to cater to the specific needs and characteristics of each domain. By examining the distribution of papers across these domains, we aim to identify trends, commonalities, and domain-specific considerations that influence the choice and effectiveness of feature selection methods. This analysis will provide a compre-

hensive overview of the diverse applications of feature selection and shed light on the domain-specific challenges and advancements in this field.

1. **Data selection in the health field**

   In the health domain, feature selection plays a critical role in various applications, such as physical activity analysis, cancer detection, neuroimaging datasets, and medical data processing. A systematic review of feature selection in physical activity analysis revealed that most studies shortlist features based on previous research, often focusing on statistical features [24]. They also point out that feature selection is about trading-off speed for robustness: with too much data, one spends too much time; with too little, the resulting model is not robust enough. Principal Component Analysis (PCA) and correlation feature selection emerged as commonly used techniques.

   In the context of cancer detection, feature selection methods like Genetic Algorithms, Particle Swarm Optimization, and Information Gain were compared, with Particle Swarm Optimization (PSO) showing superior performance [34].

   Neuroimaging datasets demonstrated varying impacts of feature selection methods, with some cases showing substantial effects while others exhibited minimal changes. The choices of feature selection algorithms in medical data analysis, such as regularized random forest and lasso, was evaluated. The domain-specific knowledge, particularly in disease morphometry, significantly contributed to the effectiveness of feature selection [47].

   Furthermore, the selection of appropriate training data attributes, such as language, writing style, and content, was emphasized in specific use cases like braille based applications [3]. They also point out that data diversity is key in a successful machine learning pipeline.

   Overall, the health domain highlights the importance of feature selection in optimizing models, managing high-dimensional data, and improving the interpretability and efficiency of healthcare applications. Moreover, we see a bigger importance of focusing on individual problems when it comes to health data, as the use cases can get very specific, thus the quality of the predictive model must be very high. Compared to previous chapters, we see more focus on population based models and simulation approaches, as well a deeper consideration on impact and compliance with regulations.

2. **Data selection in networking**

   In the networking domain, a paper titled ”Resampling imbalanced data for network intrusion detection datasets” by [7] focuses on the challenge of network intrusion detection and the issue of imbalanced data. The paper aims to reduce the overall time of the ML pipeline iterations by addressing the imbalance in the network data related to intruder detection. To achieve this, the paper primarily emphasizes the sampling of correct data from a highly imbalanced dataset. They employ a combination of Synthetic Minority Oversampling Technique (SMOTE) and random under sampling to create a more balanced dataset for network intrusion detection.

   The paper analyzes five different forms of resampling techniques across six datasets. While the paper focuses more on resampling methods rather than specific filter,

wrapper, or embedded feature selection techniques, it offers valuable insights into handling imbalanced data in network intrusion detection tasks.

3. **Data selection in finance**

   In the financial domain, a notable paper introduces a novel approach called FSRT (Feature Selection based on Rough Sets and Tabu Search) to reduce the overall time of machine learning pipeline iterations. The FSRT algorithm combines tabu search with rough sets, using conditional entropy as a search heuristic. [48] highlight the challenges of real-world financial data, which is often noisy, abundant, and contains numerous irrelevant and misleading features.

   The rough set theory is employed to identify subsets of features that can effectively classify or model the dataset without information loss, aiming to remove redundant or irrelevant attributes. The proposed FSRT algorithm operates as a filter method for feature selection, leveraging tabu search and rough set principles. While the paper delves into mathematical frameworks and theorems, it provides limited intuitive explanations of the algorithms. Nonetheless, the FSRT approach contributes to feature selection in credit scoring for the financial domain, showcasing the potential of combining tabu search and rough sets for efficient data reduction and improved model performance.

   Similar to health, the financial domain can get heavily regulated by authorities when using AI techniques to predict user information such as credit score or anomalies in credit card usage. Having only one paper focusing on this domain signals that there is still a gap in research about data reduction approaches in this domain, that can be further explored.

4. **Data selection in movie reviews**

   In the domain of movie reviews, two papers shed light on the impact of data pruning on machine learning algorithm performance. [36] investigate the sensitivity of different algorithms to iterative data pruning, aiming to reduce the overall time of the ML pipeline iterations. By iteratively removing samples based on a predefined metric, such as starting with reviews from movies with the lowest reviewer count, they gradually prune the dataset. Surprisingly, the paper reveals that pruning the dataset does not significantly influence the performance of the models, and algorithms that perform well on the unpruned dataset also exhibit good performance on the pruned dataset.

   [36] highlight the difference between dataset selection and dataset cleaning, emphasizing the uneven distribution of classes (imbalance) as a limitation. While the paper lacks depth and primarily focuses on comparing the impact of data pruning on model accuracy using the IMDB movie rating dataset, it provides insights into the importance of dataset pruning for improved ML pipeline efficiency. These papers primarily employ data pruning techniques rather than specific filter, wrapper, or embedded feature selection methods.

   There is a pattern in machine learning where new algorithms and approaches are tested on toy datasets from the movie review domain. However, having only 2 shallow papers that investigate this might signal either a lack of interest in improving the quality of such datasets or a gap in research in this particular domain.

5. **Domain agnostic approaches**

   The agnostic approaches chapter covers domain-independent feature selection. The goal is to build a set of high-contribution and informative data points, although using more data improves performance but increases training time and costs.

   Various methods are proposed, including core data construction, uncertainty sampling, and hard negative mining, measuring the importance of learning samples, as well as the contribution of each feature. In those studies, the stability of feature selection algorithms is highlighted while the impact of feature selection on time reduction, percentage of selected features, and classification accuracy is examined. The domain-agnostic papers generally agree that no single feature selection algorithm is optimal for all datasets and suggests the use of hybrid algorithms and trial and error.

   All papers have the goal of building better datasets and improving existing ones. However, the complexity of feature selection is an acknowledged NP-hard problem, and the impact of sample size on the complexity of feature selection algorithms is considered.

   Overall, the domain agnostic approaches chapter provides insights into feature selection methods that can be applied across domains, emphasizing the need for trade-offs between accuracy, efficiency, and interpretability in selecting relevant features for machine learning tasks. It is true that building algorithms that behave as one-size-fits-all are challenging and that there exists a trade-off between having specific methods that solve niche problems and generic approaches that are built once and have a good enough performance on a wider variety of use cases.

4.11. Secondary publications

The goal of this study is to systematically survey a list of papers on the topic and provide a discussion around patterns, commonalities, as well as research gaps through the field. With the aim of providing more context to this SLR and have some background studies analyzed, this section provides a summary of the key points in those reviews identified.

1. **Importance of Feature Selection**:

   Feature selection is crucial for reducing the dimensionality of data and improving the efficiency of machine learning pipelines. It helps in identifying the most relevant and non-redundant features, reducing noise, and enhancing model interpretability [27]. Also, feature selection contributes to knowledge discovery by providing informative insights to researchers.

2. **Available Feature Selection Methods**:

   The reviews discuss various feature selection techniques, including filter, wrapper, and embedded methods. Filter methods involve ranking or statistical measures to select relevant features. Wrapper methods treat the model as a black box and use searching heuristics. Embedded methods integrate feature selection with the learning algorithm itself.

3. **Impact on ML Pipeline Time**:

   Feature selection can reduce the overall time of ML pipeline iterations by providing an optimal subset of features for training. Many referenced papers stress out that selecting a small yet representative subset is important for real-time ML systems [24].

4. **Dataset Reduction**:

   Feature selection attempts to reduce the dataset by filtering out irrelevant or redundant features. Techniques like PCA, correlation feature selection, and L1 norm constraints are commonly used to reduce the dimensionality of data.

5. **Preferred Methods**:

   Filter methods, such as PCA, correlation feature selection, and statistical measures, are widely employed in the reviewed studies. Wrapper and embedded methods are also mentioned but are less commonly used [27, 24].

6. **Stability and Robustness**:

   The stability of feature selection algorithms, i.e., their ability to produce similar features under varying training data, is highlighted. Stable algorithms are important to ensure consistent feature selection results and avoid wrong conclusions. Regularization techniques are suggested to address issues of small input data changes leading to significant output changes [25, 24].

7. **Hybrid Approaches and Trial-and-Error**:

   Hybrid algorithms are often preferred due to the lack of a single algorithm that suits all datasets. Trial-and-error is emphasized as an essential approach to adapt feature selection to diverse datasets [30].

8. **Performance Evaluation and Metrics**:

   The reviews consider metrics such as classification accuracy, training time, and model interpretability to evaluate the quality of feature selection algorithms. They compare and analyze the performance of different techniques, including stability measures and subset selection for regression (SSR) methods [27].

Overall, the literature reviews highlight the significance of feature selection in different domains and provide insights into the available methods, their impact on ML pipeline time, and dataset reduction. They underscore the importance of selecting relevant features, reducing noise, and ensuring stability and robustness of feature selection algorithms. Hybrid approaches and trial-and-error are recommended to address the diversity of datasets, and various metrics are employed to evaluate the performance of feature selection techniques.

4.12. Common Patterns in Data Selection

As a summary of the SLR and as a build up of the work we are presenting in the next chapter, this section will include commonalities, patterns and future research directions when it comes to reducing datasets, collect better data, and, in general, build better and more efficient machine learning pipelines.

1. **Common Approaches**

   Across the paper pool of the SLR, we have seen many times heuristics and statistical approaches that assess the importance of each feature without taking into account the model. Those are mostly correlation and statistical-based methods. While efficient, having the system as a whole providing input to the optimization flow is valuable, especially when designing domain-specific machine learning products. Some papers discuss wrapper methods that take into account the results of training the model, that make the most out of the feedback received from it. However, those are less computationally efficient. Some of the papers are also considering less easy to quantify factors such as cost of acquisition, relevance, and reliability.

2. **Common Issues**

   Most papers agree that data selection is an optimization challenge, with the time complexity growing the more one tries to aim for global optimums. Some also mention that this can become an NP-hard problem. Also, it is agreed that good features are being overlooked using more shallow Filter models.

   Some papers are using surrogate models that act as a faster main model to speed up wrapper feature selection methods. This might compromise the quality of the wrapper methods that use the full model, but still bring an advantage over full-filter approaches, that discard what the models have to say completely.

3. **Common best practices**

   After reviewing the papers on the SLR, it is clear that using a simple approach like random sampling or basic statistical measures does not compete with other more advanced methods. It is becoming more and more clear that incorporating domain knowledge and business understanding in the data selection flow is of more significance to improving more complex flows. Also, some papers are already considering the implications of picking arbitrary subsets of data when it comes to potentially biasing a model that is trained on them.

4. **Common Challenges**

   Multi-modality in feature selection and data source selection is becoming of more and more interest, as people have more complex datasets and want a model to reason given all the available data. One potential question would be how could one algorithm that has a pool of input data streams pick the best and convert the multiple modalities into a single learning flow.

   Another common challenge is incorporating business and product knowledge in the algorithm that selects the best data. While statistical measures boil down to numbers, having business and domain knowledge into a generic algorithm might come down to the multimodality discussion. After all, those answers are yet another modality fed into the model.

   As also pointed out previously, simulation and population based methods that allow for the trial and error of multiple configurations aiming to the global optimum without making the problem NP might be a good direction to look into. Having computers simulate and aim for the global optimum could be an interesting way of finding good datasets, even in a multi-modal setting.

5
Advancing feature selection performance at corporate scale

This section covers the practical work done for the thesis, within Google. In the first part of the thesis, we explored the current landscape related to feature selection and machine learning pipeline optimization. Given the learnings and the research gaps that we have identified in the literature review, we decided to apply them in a real world scenario, working with Google, a company at the forefront of technological innovation and a leading contributor in the field of machine learning and big data analytics. Given their scalable applications of Machine Learning, we think that the insights that we will draw upon applying various techniques in such a case are relevant to the end goal of the thesis, that of reducing the time of experiment iterations. Also, small and incremental improvements are more impactful at large scales, compared to measuring the impact in small, isolated scenarios. In the following sections we explain the methodology applied in the project with Google, the new approaches we came up with, and the results we got after applying this new approaches to research datasets.

5.1. Objective

To align with the goals of the thesis and to link the broad SLR, we have decided to look into feature selection further, with the aim of making machine learning flows faster. The question we originally had and that helped us drive the research was how can we improve the start of the art, build a tool with the solution we come up with, integrate and serve it at scale to Google products? A recent paper on feature selection by Yasuda et al. [52] titled ”Sequential Attention for Feature Selection”, written by people at Google pushed the SOTA in feature selection by applying an attention layer between the selection module and the actual model. We have decided to replicate and improve the SOTA set by this paper, with the aim to further reduce the number of required features and the time to find optimal feature masks while keeping the model quality at the same standards. Usually, the quality of the model is assessed by looking over the accuracy/error on the test set. While we do this, we take into account other factors, such as train time, size of optimal feature set and the size of the reduced model. We aim to look into trade-offs between accuracy and training time.

5.1.1. Sequential Attention

Yasuda et al. [52] present a method for feature selection that relies on an attention mechanism to find an optimal K dimensional boolean mask, where they start with an empty set and sequentially add the next best feature, as outlined in Figure 5.1. At

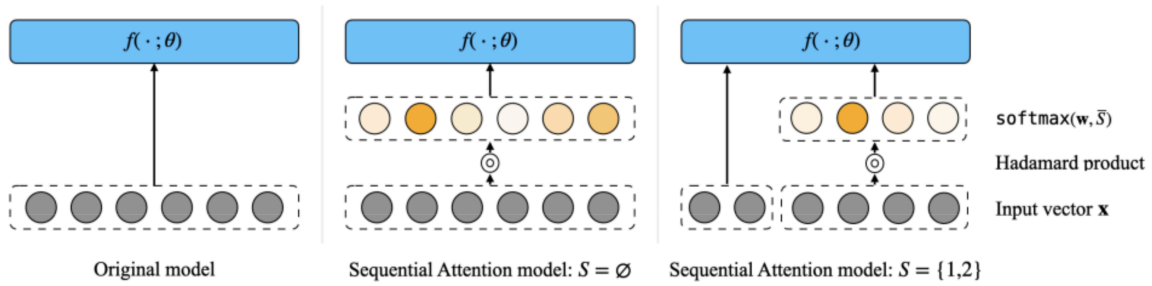

Figure 5.1: Sequential Attention Steps from an empty mask to a mask of desired size, picking the next best feature sequentially.

each iteration, a new set of features is chosen based on the highest attention logit values. The Sequential Attention module precedes the model and is actively trained using the gradients of it.

Sequential Attention holds the current SOTA on selecting a low number of features that result in a high model quality, in comparison to other approaches. Given the details in the paper, we propose two directions of improvement: having **more efficient feature selection** (ensembling and parallelization for the same results) and **increasing the quality** of their results, using similar feature subset sizes (same or lower runtime, with higher performance on the test set). For efficiency, we explored feature selection further (extending the work in the paper by looking into ways to find the same subsets faster or with fewer resources), as well as dataset and feature efficiency together, researching the impact of reducing the feature set, while randomly subsampling the dataset, to further reduce the end-to-end time, aiming for the same model performance as with the full dataset.

In [52], Yasuda et al. use a predefined size of 50 (consistent with the papers they compare their results to) for their feature subset. Our goal is to understand how much this number can be dropped in relation to the total number of features of the dataset, as well as find ways to reach a subset of a given size faster. As preliminary results show in Figure 5.2, the number of features can be significantly dropped from the predefined number of 50, while maintaining the same predictive quality on the test set of the final model. As not all datasets are built equally, the degree of redundancy can vary from one to another, and with it, the size of the optimal subset of features.

Looking into ways to reduce the time for finding such a mask, a recent paper on ensembling [16] shows that using multiple weak predictors to vote for the final prediction improves their quality. Those predictors, having different random initializations, learn different patterns that might not be predictive enough independently. However, as results will show later, ensembling the predictions yields stability and results comparable to or exceeding the results of a single learner. Given this, we focus on understanding how a multi-expert approach can improve the SOTA set by Yasuda et al. [52].

5.1.2. Improving Machine Learning Iterations

As we have seen in the SLR, relevant literature focuses on picking the features that have the best correlation with the target variable. Usually, the goal is to maximize the accuracy of the model or to draw insights on the dataset. This is important, however,

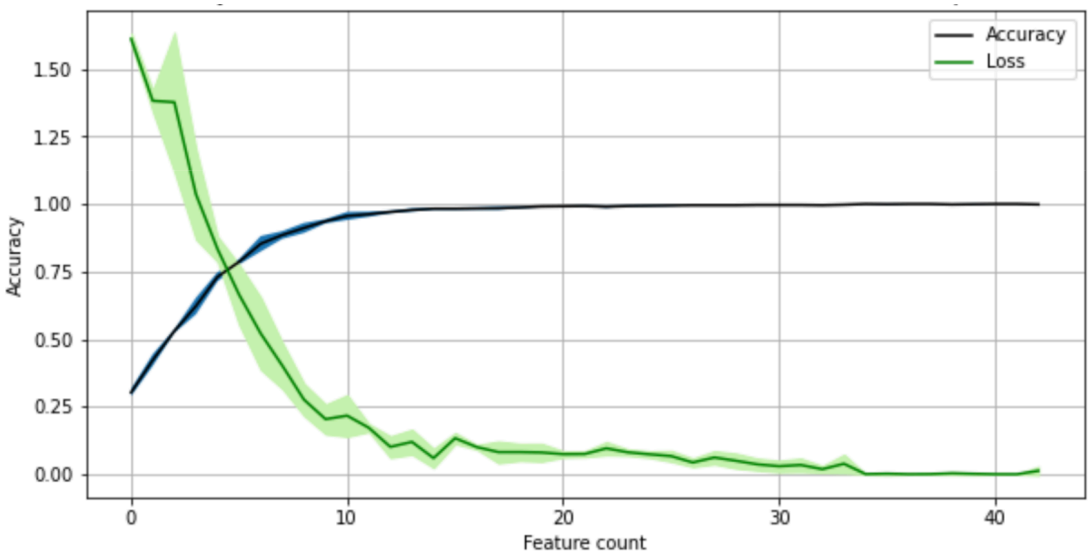

Figure 5.2: Accuracy with K selected features: SA Selection (Mice Protein). Average across 5 runs.

looking into how machine learning flows can be optimized looking into memory, speed, and resource usage is also important. In this work we focus on both maximizing the quality of the final model (e.g. accuracy), but also looking into ways to reduce training and inference time, as well as model size.

Prior literature focuses on optimizing for performance (e.g. test set accuracy). There is a gap in understanding the tradeoff between model simplicity, training/inference time, and model performance, that we aim to address in the following sections. We run experiments to understand whether there are situations where it is reasonable to make a compromise between model quality for a drastic reduction in a number of features and lower runtime. Also, we look into ways to find the optimal number of features instead of predefining it and test whether the value used by the Sequential Attention paper can be further dropped on certain datasets. After the experiments, we describe a tool that looks for optimal feature masks and provides insights on how a model performs on certain feature sizes.

5.2. Methodology

Having the Sequential Attention paper as the baseline, we plan to build on top of their open source code, to understand how we can leverage ensembling to achieve the same quality and reduce the time, as well as understanding how many of the features can be dropped, without much performance loss. Also, we aim to understand how feature selection interacts with datapoint selection, and whether finding the optimal mask can be achieved with a considerably lower amount of data.

Most of the times it is complicated to optimize all objectives at once. Usually, there is a priority of one objective over the other. Thus, we split the research directions into 2 categories, based on the improvement target:

1. **Quality Improvements (QI)**: focusing on improving the quality of the model given a fixed feature set size.

2. **Efficiency improvements (EI)**: focusing on reducing the end-to-end time and the feature set size.

Below, we outline the main directions initially considered, together with the intuition and the goal for each. Later in the thesis, we expand on the relevant directions presenting the findings and insights.

5.2.1. Research directions considered

- **[QI + EI] Ensemble sequential attention**

  The first direction we started looking into is how we can leverage the knowledge of multiple experts and come up with better results consistently. Other than benefiting from the multi-expert approach, we believe that running them in parallel would keep the same end-to-end time, while increasing the quality of the results. There are multiple ways one can use an ensemble to vote for the best output. We decided to run Sequential Attention with an ensemble of size K (usually in the order of tens), where each member would produce a mask of size M. We would then run majority voting and select the top M highest voted. Having the mask, we would use it to train a sparse model until convergence and compare the results to the benchmark.

  The main reason for using an ensemble on top of Sequential Attention is the inherent randomness of the algorithm, which can miss important features if a single runner is used. With voting, there is an increased chance that the algorithm will produce better results despite the random initialization (Figure 5.3 shows the reduced variance of the Ensemble/Intermediate Sequential Attention, highlighting that in comparison to Simple Sequential Attention, it is more stable). Having a multi-expert approach can also validate the importance of some specific features. One can identify the highly predictive features by analyzing the voting heatmap. The more a feature is selected by the pool of models, the higher the likelihood it is an important one.

  Improvement goals:

  - **QI**: increase quality, stability and achieve consistent better results.
  - **EI**: run workers in parallel and reduce the end-to-end time.

- **[QI + EQ] Sequential attention with feature batches**

  Similar to the previous plan on simple ensembling, we aim to select the optimal D features out of N, with the feature set split into K equal parts, one for each worker. Each worker is expected to select approximately D/K features from their subset. Once the ensembling is done, the algorithm will come up with the final set of features. Instead of voting, the final step would be to put together the feature subsets of every batch. The goal is to increase parallelization on the previous ensemble method, as all the ensemble members can, in parallel, process their subset.

  Improvement goals:

  - **QI**: have workers specialize in a feature batch instead of focusing on the complete set.
  - **EI**: run each batch in parallel and gather results.

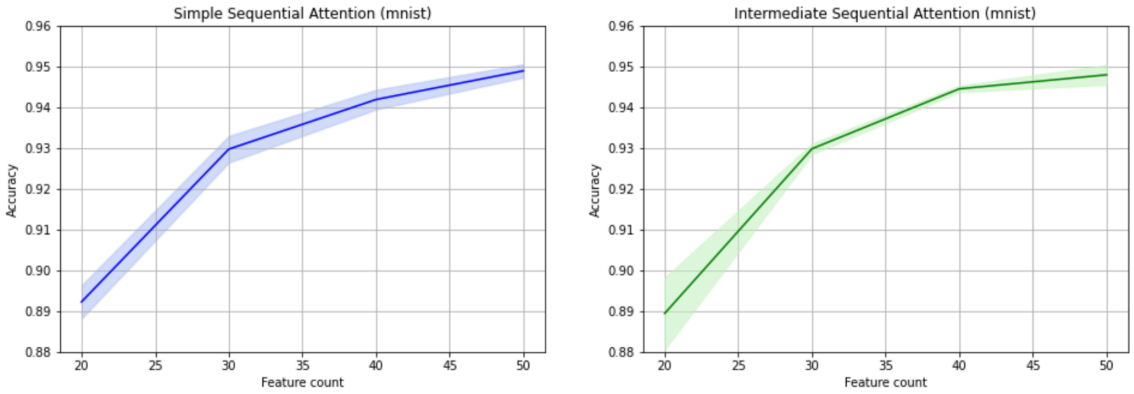

Figure 5.3: Simple SA VS Intermediate SA. When using an ensemble of multiple workers voting for the final mask, the accuracy is more stable. Average across 5 runs.

- **[QI] Multi-headed attention embedding**

  The Sequential Attention paper uses a simple Hadamard product to select features, turning them on/off based on the value of the attention logit. However, this does not consider feature interactions. In the Attention is All You Need paper [46], the authors propose the transformer, a seq-to-seq specific architecture, with an attention mechanism that takes into account the relations between different tokens in a sequence. Despite the fact that the use case detailed here does not imply a sequential data model, the same logic can be applied to reduce the dimensionality of the dataset, while taking into account the interactions between features.

  Instead of using the Hadamard product to switch features on/off, we attempt to use a multi-headed self attention layer. In this case, the output of the attention layer would not be a mask (a boolean vector), but an embedding representing the feature set. Based on the size of the embedding « |featureset|, this can behave both like a feature selection and feature engineering step, creating a smaller and more interaction aware embedding.

  Improvement goals:

  - **QI**: learn from the feature-to-feature interaction and produce better representations.

- **[QI + EI] Intermediate sequential attention**

  As an expansion to the initial ensemble approach that can already be parallelized, we looked into modifying the sequential attention flow to use K workers. Instead of having a pool of K models, each working independently and producing a final mask, the intermediate algorithm would synchronize the models periodically, have them vote for the next best few features, and maintain a common prior. The goal is to enhance diversity while also capturing the most important features in the data.

  As the experiments will show, having an ensemble of models doing Sequential Attention independently leads to picking features that are redundant together. Having a single Sequential Attention flow with K workers that sync on the selected features

periodically reduces the chance of redundancy in the final mask, while allowing workers to still work in parallel.

Improvement goals:

- **QI**: reduce the influence of random initialization of ensembling individual workers.
- **EI**: run each worker in parallel, and gather the results at each intermediate step.

5.3. Methods and Experiments

Based on the ideas outlined in the previous chapter, we expand the current implementation of Yasuda et al. [52] and we ran extensive experiments to validate our assumptions and understand future directions and opportunities to optimize the methods used. The goal was to understand how the whole machine learning flow can be sped up while either keeping the same quality of the final models or having an acceptable tradeoff between time-saving and the metric we are optimizing for (e.g. accuracy).

5.3.1. Datasets setup

Since the Sequential Attention for Feature Selection is our baseline, we largely used the datasets they provided results for, and the same open-source code for the Sequential Attention module, together with the same hyperparameters to find a mask and train the final model.

Table 5.1, and table 5.2 present the datasets and hyperparameters used for running the experiments.

Table 5.1: Data hyperparameters

Dataset name	Data Type	ML task	Batch size	Validation Ratio
Mnist	Image	Classification	349	12.5%
Mnist Fashion	Image	Classification	391	12.5%
Activity	Tabular	Classification	183	12.5%
Mice protein	Tabular	Classification	16	12.5%

Table 5.2: Model hyperparameters

Dataset name	Learning rate	Decay Rate
Mnist	$6e-3$	0.37
Mnist Fashion	0.4	0.84
Activity	$1e-5$	1.00
Mice protein	1.0	0.63

5.3.2. Improving the Sequential attention subset size

The first step in building on top of the Sequential Attention paper was to replicate it and study its stability, as well as the ability to decrease the number of features lower than 50 without a considerable drop in performance. We ran experiments where Sequential Attention would find the best feature mask for each size in [1 .. 50]. As we've previously seen in Figure 5.2, in the case of the Mice protein dataset, not only is Sequential Attention stable across different random runs, but the same test set accuracy can be achieved with as low as 18 features, a 64% decrease from the benchmark 50 features and a 77% decrease from the initial 77 features. The results were interesting as the number of features for optimal performance can be drastically reduced from 50. Here are the experiments. This signals that even if Yasuda et al. [52] used 50 features in their paper, there is still redundancy in some of the datasets used that allow for a much smaller feature mask, without affecting quality.

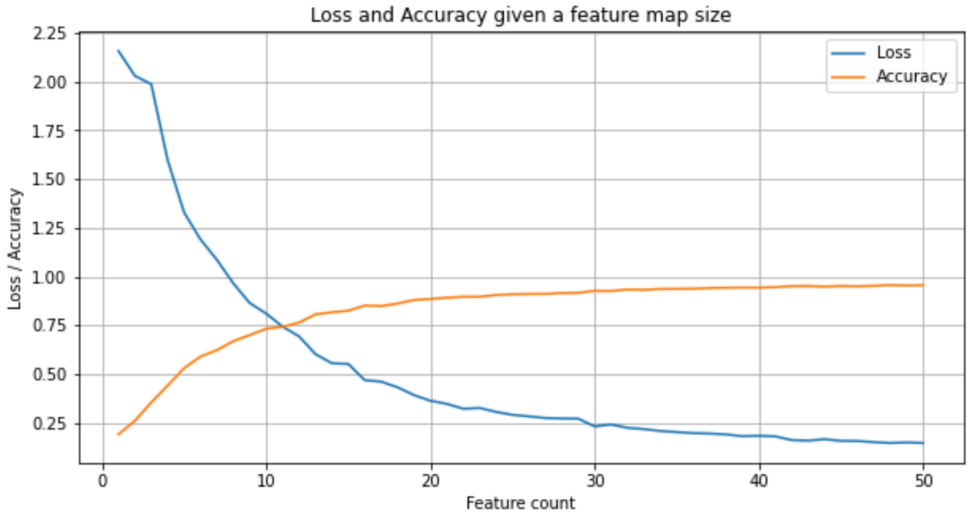

Figure 5.4: Loss and Accuracy given a feature map size (MNIST). Continuously adding features increases the accuracy of the model. Average over 5 runs.

For visual datasets, however, the redundancy exists but it's less emphasized. For MNIST, continuously adding features, always increases the accuracy of the model, as can be concluded from the accuracy plot in Figure 5.4. However, even with this information, one can still find a tradeoff between runtime/memory usage and quality. With 50 features the accuracy of the model on the test set is around 95.5%. Further reducing the number of features by 50% implies a drop in performance of only 4%, which might be acceptable in certain situations where speed and dataset reduction are more important.

In Figure 5.5, we ran Sequential Attention for a target mask size ranging from 1 to 50. As the number of target features increases, there is a tendency for the Sequential Attention module to pick features that are closer to the center of the image rather than toward the borders. This is the expected behavior since the MNIST images are centered, usually with a significant black border.

Figure 5.6 highlights a similar pattern, but for a tabular dataset. Since tabular datasets don't have a second dimension, and there is no 'center' for them, we'll refer to

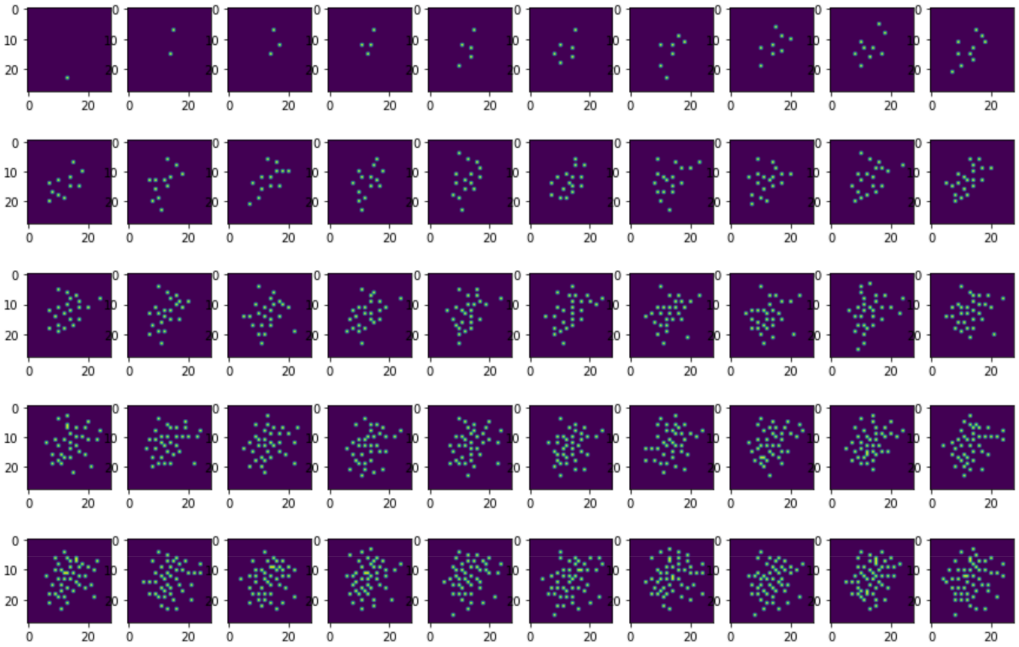

Figure 5.5: Pixel map for increasing feature set size (MNIST). Each plot represents the optimal feature mask for a specific number of features, ranging from 1 to 50.

the features as being cells on a single grid. While in Figure 5.5 the significant features of the MNIST dataset are towards the center of the image, in Figure 5.6 there is a tendency to pick features from the right part of the grid, highlighting that there might be significant redundancy in the features on the left side relative to this prediction task.

For the Activity dataset, the same situation occurs, where picking a smaller feature set size yields good results. In that case, one can take more advantage of the trade off between a drastically smaller mask for a small drop in performance. As can be seen in Figure 5.7, one could reduce the features by 60% from the benchmark, while dropping the performance by less than 2%.

For the Activity dataset as well, Sequential Attention based feature selection outperforms random selection for masks of at least 4 features (Figure 5.8).

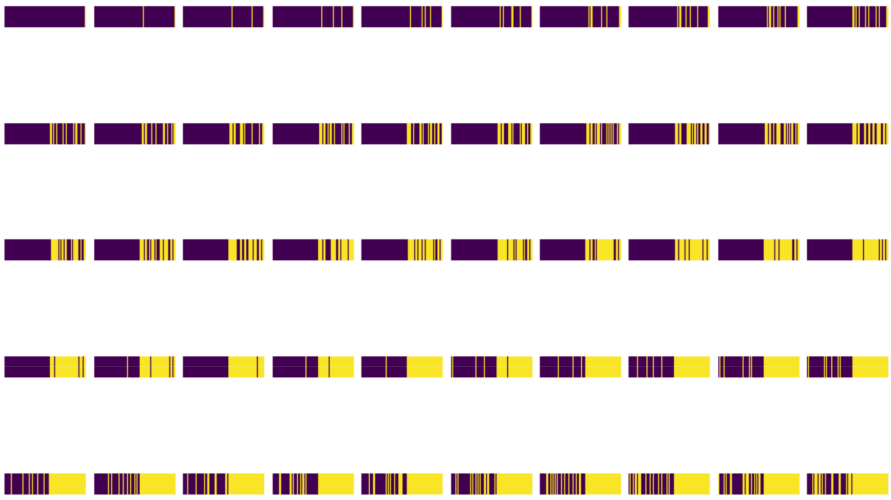

Figure 5.6: Feature map for increasing feature set size (Activity). Each plot represents the optimal feature mask for a specific number of features ranging from 1 to 50.

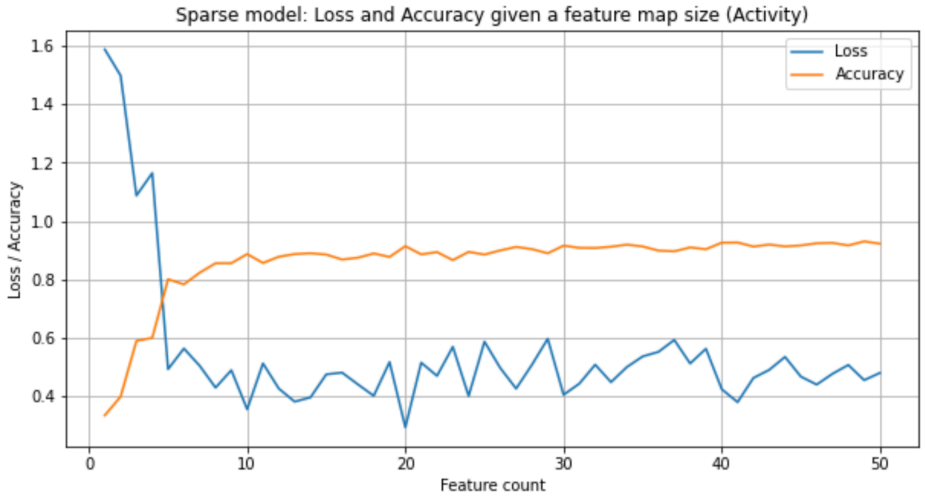

Figure 5.7: Loss and Accuracy given a feature map size (Activity). The difference in accuracy between 50 and 30 features is less than 2%. Average across 5 runs.

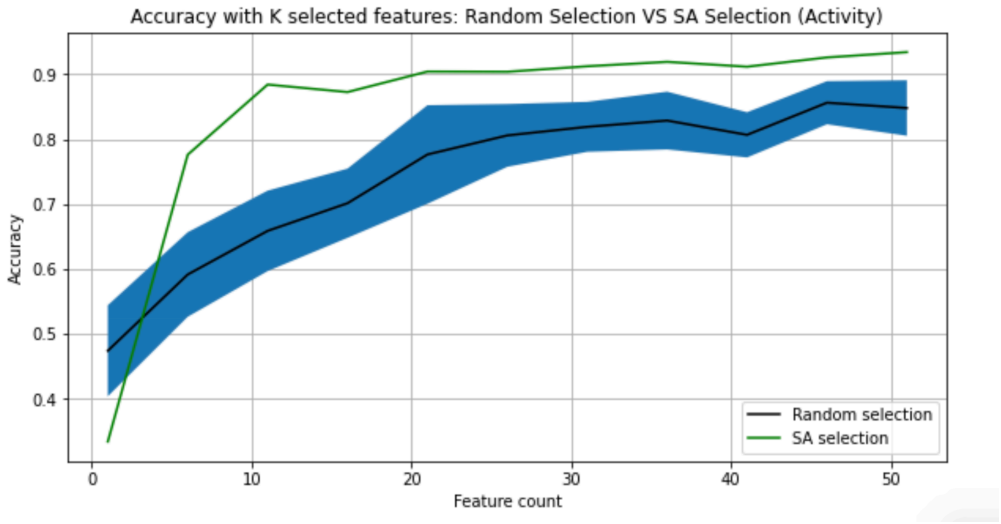

Figure 5.8: Accuracy with K selected features (Activity), single Sequential Attention run VS average across 5 random selection runs.

5.3.3. Ensemble sequential attention

To start with this approach, we ran multiple ensemble sequential attention models and compared the results to the ones of single sequential attention models, on the MNIST and fashion MNIST datasets. As a first approach, we used multiple workers that would individually come up with a mask of size K, gather their results and pick the most popular global K features as the final mask. In Figure 5.9, we can see the commonly used features that maximize accuracy (The Common mask), a heatmap with feature popularity in the case of voting methods, and a few data samples for reference.

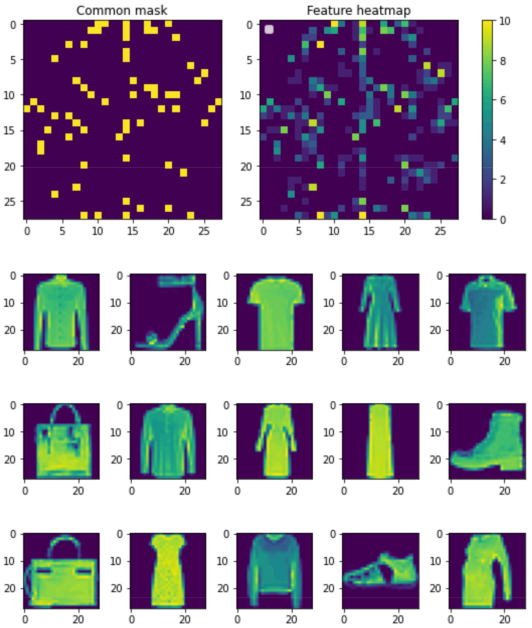

Figure 5.9: Mnist heatmap and example images.

A common issue in feature selection by multiple experts without coordination is the likelihood of choosing redundant features. For example, in a dataset with features [A, B, C, D, Y], A and B are both strong predictors for the independent variable Y, significantly affecting accuracy, while C and D are more marginal, capturing key outliers. When an ensemble aims for a feature pair, the high ranking of A and B often leads to their inclusion in the final set. However, this may not be optimal; models trained on combinations like [A, C], [A, D], [B, C], or [B, D] might outperform those trained on [A, B] due to A's and B's redundancy when they are considered together. This phenomenon is particularly noticeable in datasets like MNIST, where visual analysis shows that an ensemble method tends to favor too many similar central features, as seen in the comparison between simple SA and ensemble-generated masks in Figure 5.10. On the left side is the mask generated by just one sequential attention runner, and is evenly distributed, without many clusters of features, enhancing diversity. On the right side, there is the output of an ensemble of 10 sequential attention runners, voting for the final mask. It is noticeable that there are clusters formed in the center of the image (the redundant together features), while there is not much diversity towards the borders.

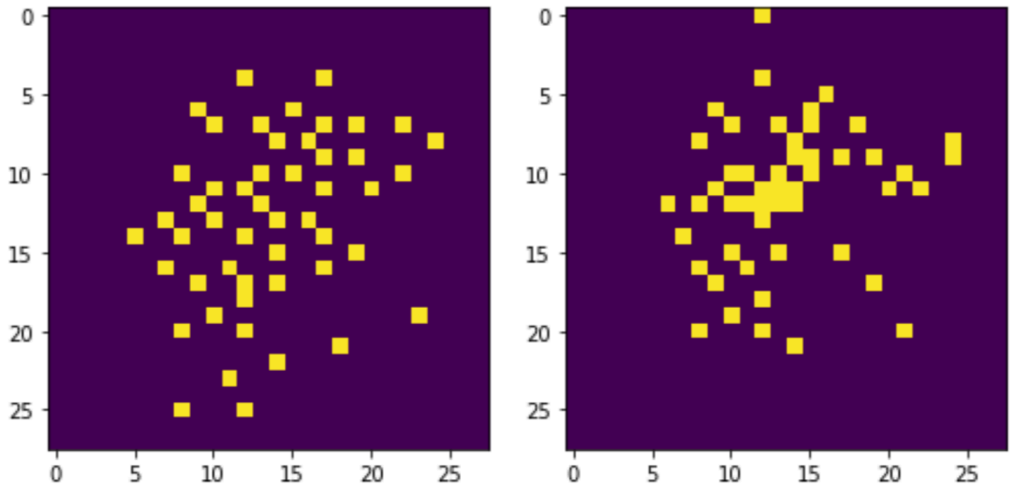

Figure 5.10: Redundancy and lack of diversity in ensemble voting. On the left is the mask resulted from a single Sequential Attention layer, while on the right there is the mask agreed on by an ensemble of Sequential Attention workers. In the later it is clear that many features are redundantly picked together (towards the center of the image), hindering the diversity towards the borders.

The accuracy difference between the 2 masks in Figure 5.10 can be as high as 2% on the test set, due to the redundancy problem. To mitigate this, we used the same ensemble and voting method, but instead of all the models having selection access to all the features, we randomly masked a large number. The masking factor ranges from 0 to 1, with 0 meaning that the model sees all the features and 1 meaning that the model sees no feature.

The more features (pixels in the case of MNIST) are randomly masked out, the more the ensemble tends to pick a more diverse set, lowering the chance of picking features that are redundant together, as it happens when all the models have access to all the features. In Figure 5.11, 70% (left) and 85% (right) of the features are masked.

It is visible how the vote heatmap covers a more diverse range of pixels. However, none of them is able to create feature masks that come close to the accuracy of the feature mask generated using Sequential Attention only. This also enables us to make the statement that randomly masking out features might not yield better results.

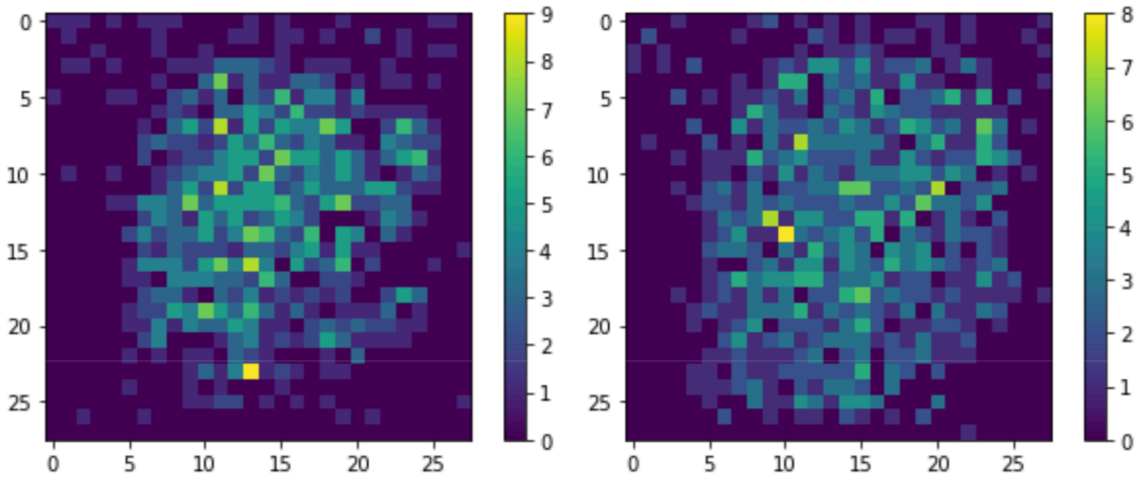

Figure 5.11: Diversity in masked ensembling. For those feature vote heatmaps, a random number (70% left, 85% right) of features was masked out for each member of the ensemble, with the goal of enhancing diversity.

5.3.4. Sequential attention with feature batches

As outlined before, we attempted to run Sequential Attention on disjoint batches of the feature space, and concatenate the results that are then used to train a sparse model. However, this implies that every batch is weighted equally, both in terms of the importance of features as well as the number of features that should be picked from it (Figure 5.12). As results indicate (especially the plots on feature selection for every iteration from 1 to 50), this is not necessarily the case.

With a mask generated by putting together the best features from every grid the model achieves 90% accuracy on the test set for MNIST, far from the SOTA (with reduced feature count) of 96%. One potential future direction can be looking into how the weight of each region (and the number of features to be selected from it) can be learned through gradient descent, so the model can learn what regions to pick features from. However, with this approach, we expect the model will quickly “forget” about the concept of regions, and converge to the result of Sequential Attention on the full featureset, where no regions are defined. When running on tabular data, the results are slightly better, possibly due to the non-localized nature of the tabular dataset. In this case, even a shuffled version of the feature space would resemble the same data, while for images it would not hold true. For Mice protein, the grid based approach equals the SOTA at 100% test accuracy, while in the Activity dataset it reaches 92.4% (SOTA 93%). However those are isolated cases where even if one picks an equal number of features from each equally sized grid, the most globally important features are also likely to be picked.

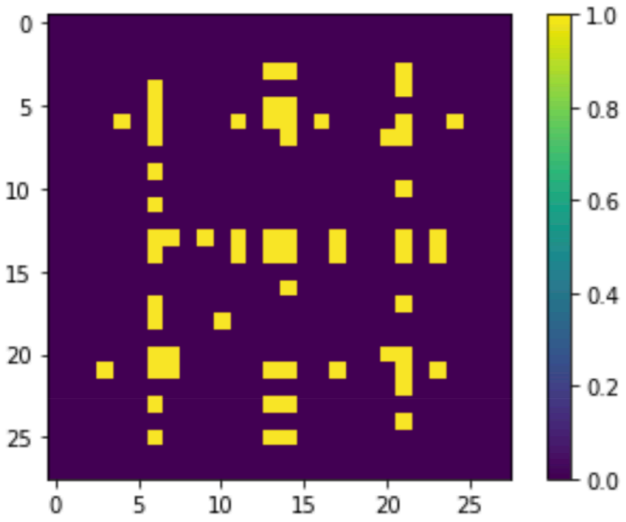

Figure 5.12: Final feature subset with feature batches.

5.3.5. Intermediate Sequential Attention

Previously, we have looked into training multiple Sequential Attention models individually that vote on the top K features once all of them end the training. The mask that all the models agree on performs worse than using a single Sequential Model. In fact, in some cases it performs worse than the average performance of the models in the ensemble, leading to wasted resources training all the K models. One of the problems we identified is the agreement on features thats are redundant together, at the expense of not picking the features that would capture more variance in the data. This is caused by the lack of synchronization in the ensemble process combined with random initialization of each model, that leads to generating, at each step, a next best feature with different priors.

To mitigate this issue, we introduce Intermediate Sequential Attention, a flow that aims to combine the quality of simple Sequential Attention with the value added by the multi-expert approach of ensembling. The algorithm is as follows:

- Having a dataset with N features,
- and a budget of M runs, each with a specified target feature count: $[m_1, m_2, ...m_3], with\ m_i < m_{i+1}$
- compute the best mask for each of those targets,
- using an ensemble of K models that vote for a common mask at each m_i.

The sequence of steps from above is formalized in Algorithm 1.

You might notice in the training of each individual model that the number of epochs is scaled to the size of the ensemble. The hypothesis here is that we want to keep the same total compute time, but parallelize it. If a normal, single Sequential Attention model would achieve performance X in E epochs, we want the ensemble to achieve the same quality, in parallel, in a total end-to-end time of approximately E / ensemble_size epochs. The quality of each individual model will be poor compared to the quality of the individual runner, but through the voting mechanism the overall performance of the ensemble is comparable, in a fraction of the time. To draw a parallel

to multiheaded attention, each ensemble member can be seen as one attention head. Given different random initializations, each member picks a set of features, in a shorter timeframe, based on the prior that they hold from the initial weights. Each member might pick a subset of the features that might not be entirely relevant to the final use case, but having multiple models vote for the set of features improves the quality of the ensemble.

Algorithm 1 Get Masks Algorithm

```
1:  function GetMasks(target_counts, ensemble_size)
2:      mask ← []                                   ▷ The prior for all the models
3:      for t = 0 to |target_counts| do
4:          target_count ← target_counts[t]
5:          features_needed ← target_count − |mask|
6:          model_pool ← []
7:          for m = 0 to ensemble_size do
8:              model ← Model(prior: mask, select_feature_count: features_needed)
9:              model.train(epochs: epochs_select/ensemble_size)
10:             model_pool ← model_pool + model
11:         end for
12:         feature_votes ← get_votes(model_pool)
13:         new_features ← select_top_k(feature_votes, k: features_needed)
14:         mask ← mask + new_features
15:     end for
16:     output ← []
17:     for t = 0 to |target_counts| do
18:         target_count ← target_counts[t]
19:         output ← mask[: target_count]
20:     end for
21:     return output
22: end function
```

The more fine grained the target counts are, the less redundant-together features will the ensemble members pick. If the targets are in increments of 1, the models will vote on the next best single feature and the chance of getting a low performing final mask is lower. On the other hand, picking a too large gap between the target counts (e.g. [1, 50]), would transform this flow in the simple voting algorithms from Section 5.3.3, where the models vote for the entire mask, leading to a mask that yields poor performance in the end, due to the redundancy issue.

We have also looked into generating the feature set in increments of 1, but this results in more computational resources required, since the lifespan of each ensemble worker is lower, and given that generating the worker jobs and synchronizing their results takes a non-trivial time, we have decided to focus on experiments with larger steps, since they offer a better time to accuracy ratio. Also, when each worker is tasked with picking a larger number of features (larger than 1), there is a higher chance that the voting algorithm will have a clear maximum to add to the feature set, rather than dealing with ties and randomly picking a feature in the case where each model only looks into the next single best feature.

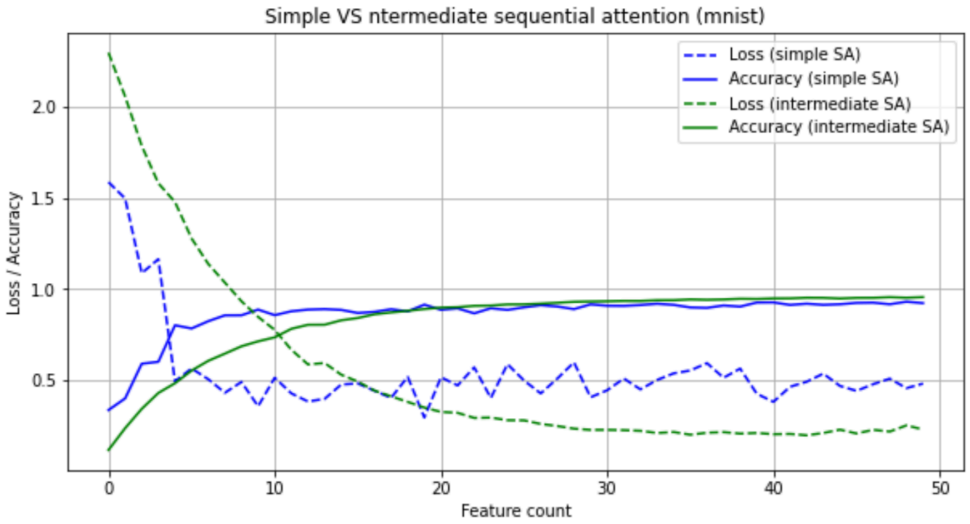

Figure 5.13: Simple VS Intermediate SA (MNIST). Average across 5 runs.

In Figure 5.13, there is a comparison between accuracy achieved with normal Sequential Attention for K features and the accuracy achieved with Intermediate SA. The first observation is that Intermediate SA is more stable than simple SA. The intuition is that Simple SA does not have any prior, and starts from a random configuration for every feature count, while Intermediate SA always builds upon the previously selected feature, hence the monotonically increasing trajectory. This also explains the lower accuracy compared to Simple SA until 16 features. Once the ensemble picks a set of initial features, the prior is the same for the whole run, while Simple SA re-evaluates the feature set every time, such that:

- $|SimpleSA[i] \cap SimpleSA[i-1]| \in [0, |SimpleSA[i-1]|]$, while
- $|IntermediateSA[i] \cap IntermediateSA[i-1]| = |IntermediateSA[i-1]|$

, restricting Intermediate SA from fixing a poorly chosen prior. However, after a certain number of features, Intermediate SA is consistently better than Simple SA.

The fact that Intermediate Sequential Attention is more stable over time makes sense given the information outlined previously. However, one interesting observation is the difference for a low number of features. Given the random initialization of both simple and intermediate sequential attention, one could expect gaps in quality, especially in a low feature regime. To better understand the impact of random initialization to intermediate sequential attention, we have built a graph to measure the variance over 5 random runs (Figure 5.14). Visually, the more features are added to the model (and the more advanced we are in the search), the more stable the voting mechanism becomes.

5.3.6. Sequential attention on reduced data

Given the positive results on further reduced feature counts, we attempted to look into both reducing the feature set size and the dataset size. To our knowledge, there is no study that looks into both directions simultaneously. Both directions are important

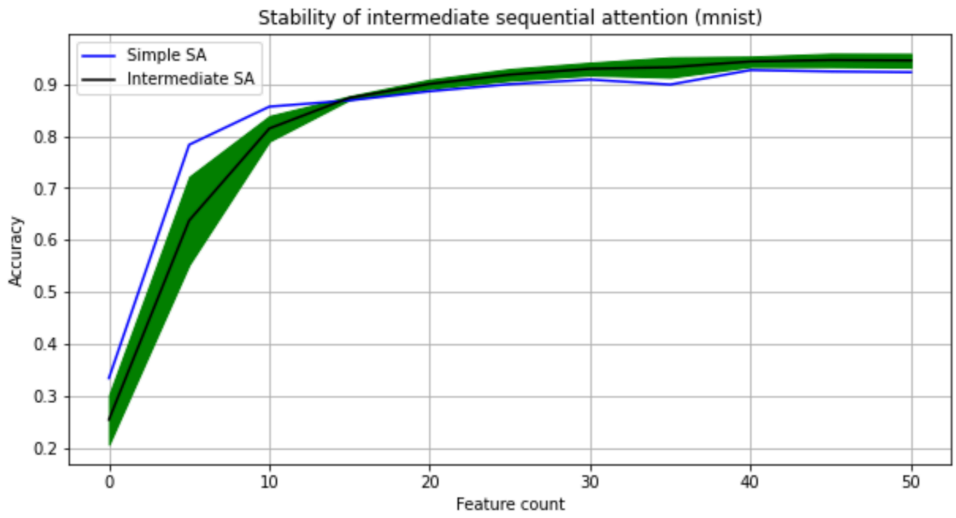

Figure 5.14: Stability of Intermediate Sequential Attention. Average across 5 runs.

individually for the purpose of time savings, thus compounding them might yield even better results.

From a time performance perspective, we aim to optimize the time it takes to find a good mask given a feature count budget. Having the mask, we will train the full model end-to-end with the whole masked dataset. Given this considerations, the following complexities hold true:

- Training a full model end-to-end: $O(N(dataset\ size) * F(feature\ set\ size))$
- Finding the mask with a fraction $X \in (0, 1)$ of the data and training the model end-to-end: $O(N' * F) + O(N * F')$, with $F' << F$ and $N' = X * N$.

Preliminary experiments on the same datasets indicate good tradeoffs in terms of final accuracy working with both low data and feature regimes (Figure 5.15).

Zooming in (Figure 5.16), we can see that having the model train on 20% of the dataset achieves performances very similar to training it on the full dataset, even with lower feature counts. This is good from 2 perspectives:

1. Running experiments faster, if the goal is to see the trend of the model, and not the peak accuracy.
2. For use cases where time is crucial and accuracy can be traded for model speed, having such plots can help decision makers pick optimal values for their data share and optimal feature set.

A similar situation occurs for image datasets as well (MNIST), as can be seen in Figure 5.17 and Figure 5.18 for a zoomed-in version.

As a next step for looking into reducing the dataset as well as the feature set, active learning can be applied for the model to decide what part of the distribution to pick data from, such that it can start with as low as 10% of the dataset, and only request more data if required. This way, the required data quantity is learned actively while also learning the best mask for a given feature set size.

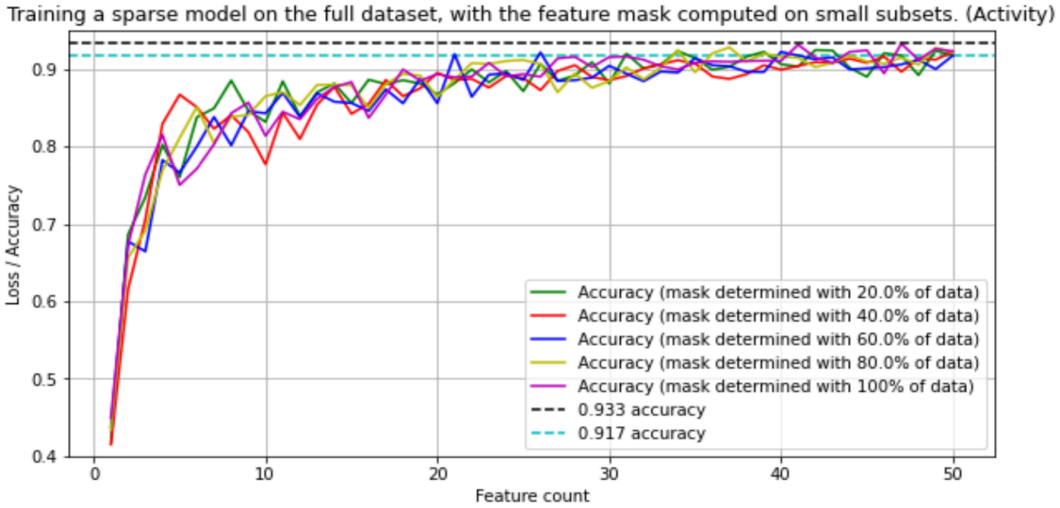

Figure 5.15: Training a sparse model on the full dataset, with the feature mask computer on small subsets (Activity). This graph depicts the relationship between feature count and model accuracy, demonstrating that models trained on subsets of data (20% to 100%) can nearly match the performance of models trained on the full dataset. It underscores the efficacy of feature selection in enhancing computational efficiency without significantly compromising accuracy.

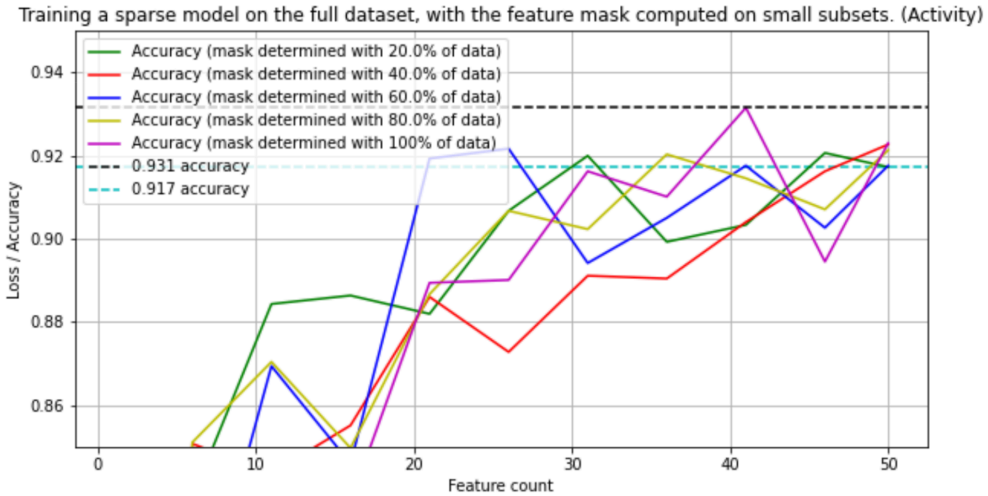

Figure 5.16: Training a sparse model on the full dataset, with the feature mask computed on small subsets (Activity). This zoomed-in graph demonstrates the precision of feature selection in machine learning, showing how models with reduced feature sets closely approach the accuracy of those trained on the entire dataset, thus reinforcing the value of efficient feature selection.

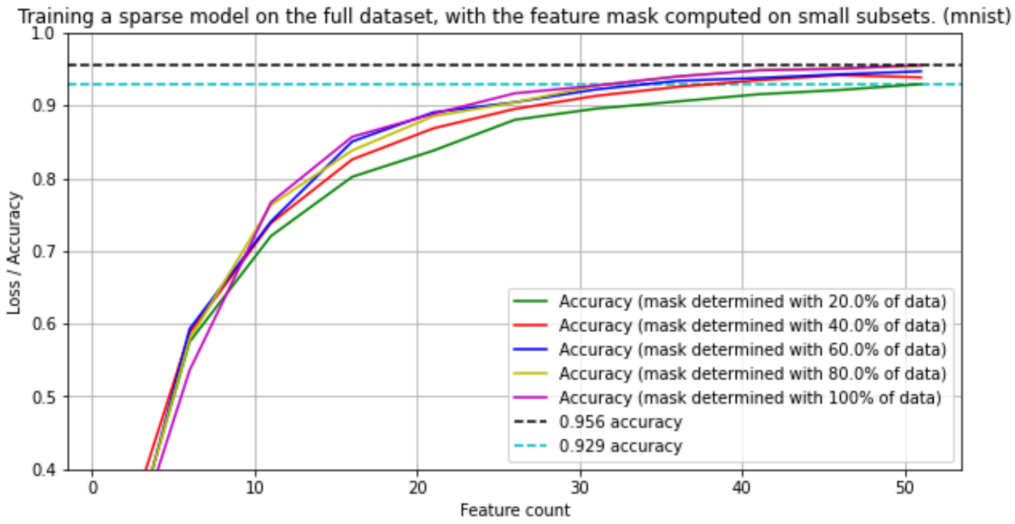

Figure 5.17: Training a sparse model on the full dataset, with the feature mask computer on small subsets (MNIST).

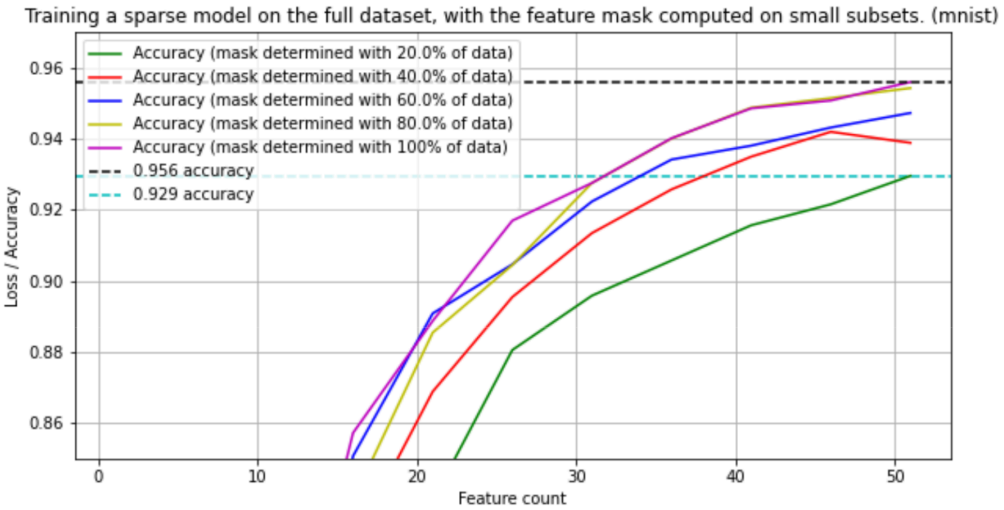

Figure 5.18: Training a sparse model on the full dataset, with the feature mask computer on small subsets (MNIST).

5.3.7. Computing mask size upper bound with SVD

All the previous experiments require the user specify the target number of features in the smaller set, introducing possible bias and potentially resulting in a mask that is either too small, thus making the model underperform, or too large, creating redundancy in the reduced feature set.

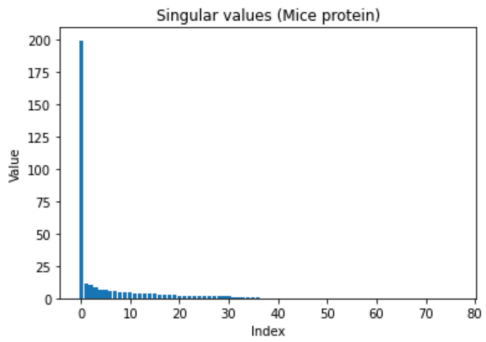

Figure 5.19: Singular values (Mice protein).

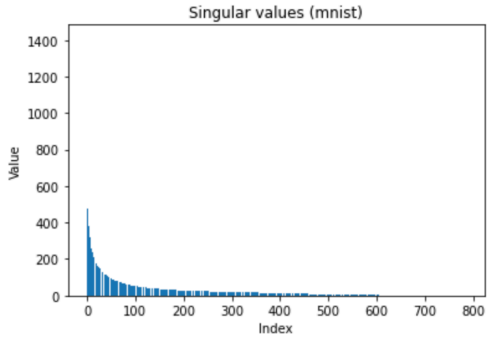

Figure 5.20: Singular values (MNIST).

Singular Value Decomposition (SVD) plays an important role in identifying an optimal feature set for various data-driven applications. It is a matrix factorization technique that decomposes a matrix into three separate entities. SVD reveals the inherent structure of the dataset and can help in choosing the optimal set of features that balances between minimality and preserving accuracy. This is achieved by selecting the size of the feature set to be the number of non-trivial singular values in the decomposition. To validate this point, we ran experiments on both MNIST and Mice protein, and as Figure 5.19 and Figure 5.20 show, the numbers are representative of the results of Sequential Attention earlier in the chapter.

For Mice protein, the number of non-trivial singular values is 36 (out of 77). However, even if not all of them might be correlated to the target column, this provides an upper bound to the size of the feature subset. In the accuracy graph, it is clear that 100% test set accuracy is achieved by picking only 20 features, thus it would be a waste of resources to compute the accuracy for 40 and 50, for example. By having the 36 upper bound provided by SVD, one could only search a subset of size lower than this, potentially saving wasted compute time.

We had a similar result for MNIST, where the upper bound for the feature subset size was 600. Still higher than the number of features that already yield a good model quality ($\sim$50), however, it offers an upper bound preventing unnecessary compute in the 600–784 features count region.

It is worth noting how the singular values are inversely correlated to the accuracy given a range for the feature count, highlighting that the first 50 features increase the quality significantly, with the rest of the features only improving the quality marginally.

A next step for Singular Value Decomposition is understanding how it can be applied to large datasets that do not fit into memory. One initial heuristic is to run SVD on small batches, retrieve the number of non-trivial singular values for each, and use the median value as the support bound for the mask finding algorithm.

5.3.8. End-to-end time comparisons

We have also run end-to-end experiments to compare the approaches discussed so far to the benchmarks in the Sequential Attention paper. Given a training budget of potential feature subset sizes, we have measured the time to get the optimal masks for all the sizes, as well as the accuracy with each mask size (Figure 5.21), for all the following methods:

- Simple SA: one Sequential Attention model, responsible for generating a mask of a given size.
- Simple SA 20: Simple SA, but on a uniform 20% sample of the dataset.
- Simple SA 40: Simple SA, but on a uniform 40% sample of the dataset.
- Simple SA 60: Simple SA, but on a uniform 60% sample of the dataset.
- Intermediate SA: Intermediate Sequential Attention, as detailed in the section above.
- LLYMask: the algorithm from [28]
- GL: Group lasso according to [4]

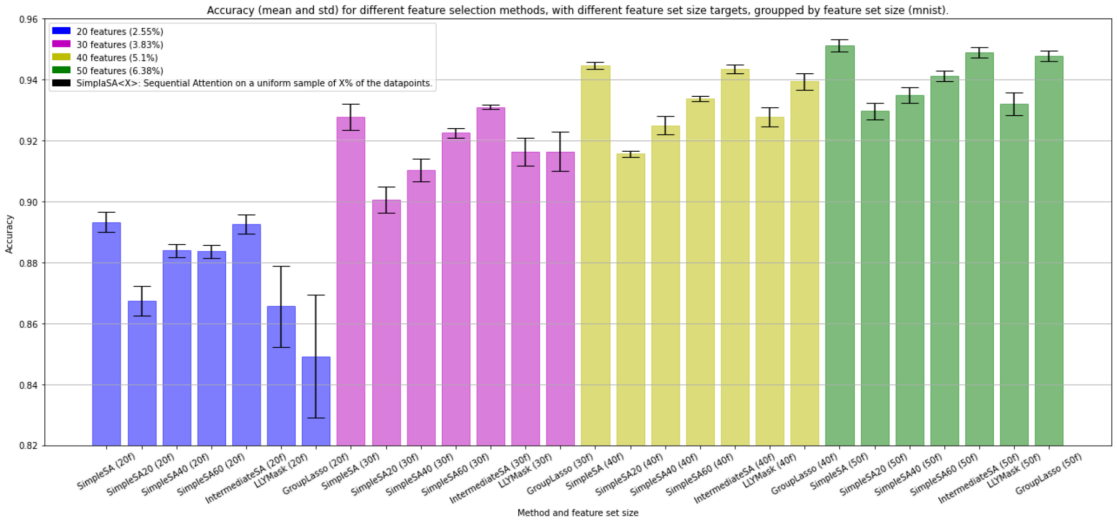

Figure 5.21: Time measurements across various methods and datasets (MNIST). Average across 5 runs.

As it is clear from the Figure 5.21, Sequential Attention and Intermediate Sequential Attention are consistently better than other approaches. Once again, the Intermediate SA has the lowest variance across all 3 feature sizes, proving the stability of ensemble methods.

Having the accuracy and time to get the mask for each of the models above, we have computed the accuracy/time ratio as a first measure of understanding the trade-off between how fast a model is and the quality of the results (Figure 5.22). However, the results in the trade-off table have to be analyzed together with the accuracy results to be conclusive on what model to use in a production scenario.

Even if Simple Sequential attention provides slightly higher accuracies than the Intermediate version, the latter has the best quality/time tradeoff, due to its parallelization abilities. Even though the total compute time is comparable to Sequential Attention,

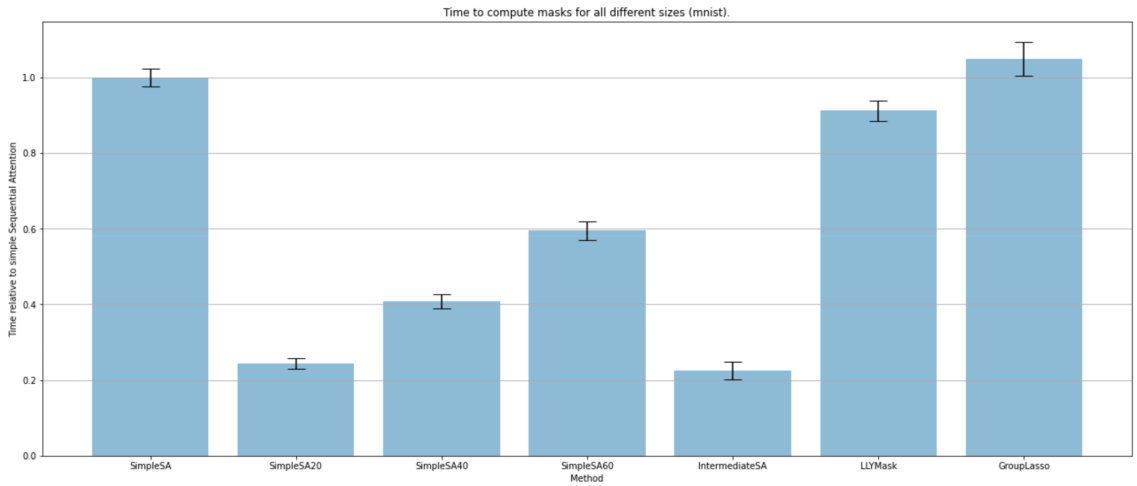

Figure 5.22: Time to compute mask for all different sizes (MNIST). Average across 5 runs.

the end-to-end time is reduced (the reduction scales with the numbers of members in the ensemble). At each epoch, K models compute the best next features in a short amount of time, then vote for the actual features that make it in the mask. At that point, the quality of each model is suboptimal, however the mask decided for after voting performs consistently well.

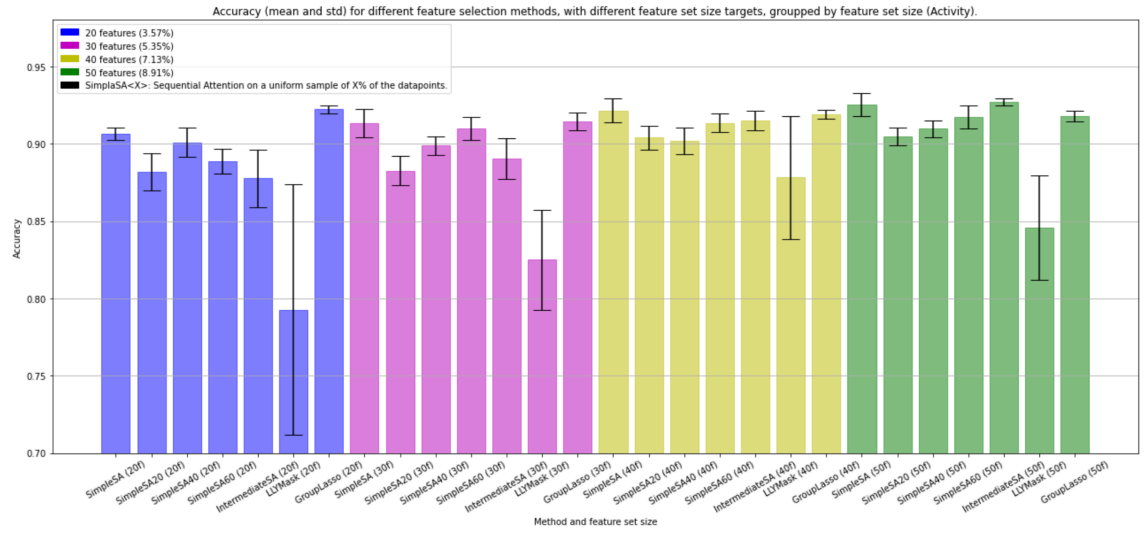

Figure 5.23: Time measurements across various methods and datasets (Activity). Average across 5 runs.

We have also run similar experiments on tabular data (Activity dataset), (Figure 5.23). For this dataset, the masks delivered by the Intermediate Sequential Attention were comparable or better than the ones from Intermediate Attention for 40 and 50 features.

In terms of time, however, the same ratio as with the experiments on mnist holds true. Due to the Intermediate Sequential Attention being parallelized, the total time to accuracy is lower for sequential attention, providing the best tradeoff (Figure 5.24).

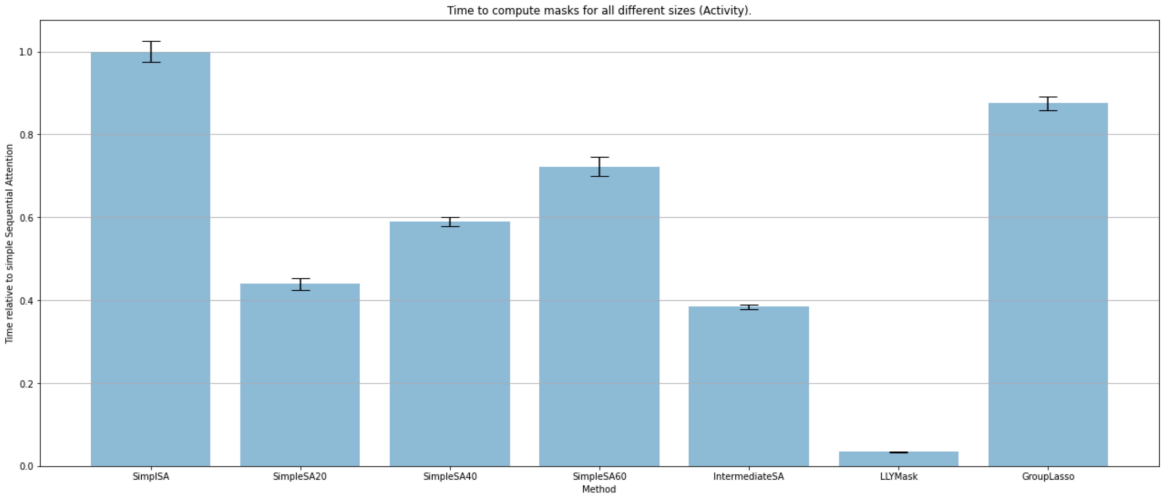

Figure 5.24: Time to compute mask for all different sizes (Activity). Average across 5 runs.

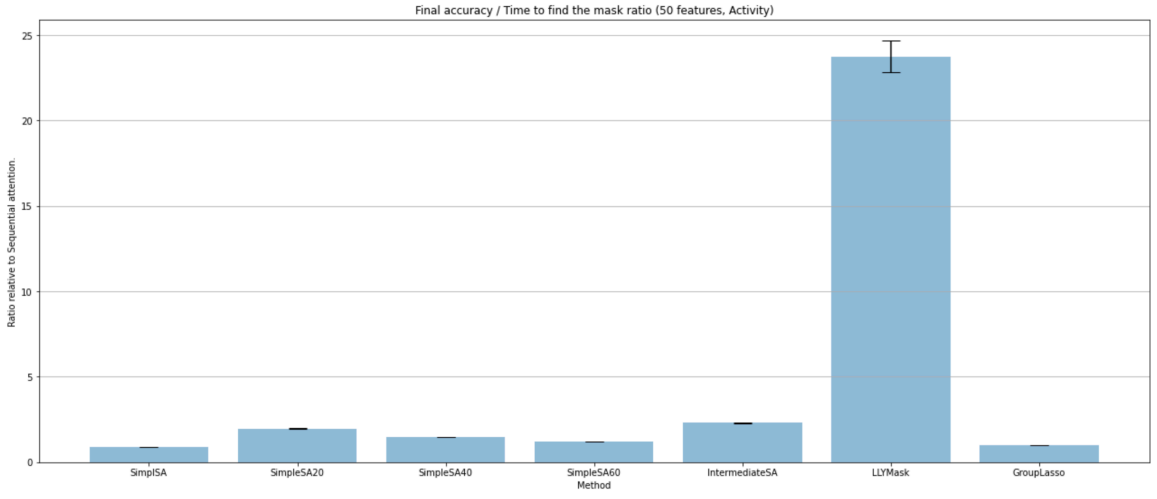

Figure 5.25: Final accuracy / Time to find the mask ratio (50 features, Activity). Average across 5 runs.

However, the time for the LLY Mask approach is very short compared to any of the other methods, and so is the accuracy. That is the reason why in Figure 5.25 on the accuracy/time ratio, it ranks very high compared to every other method. That is why those charts should be analyzed together, since sequential attention is the best method given the tradeoff between accuracy loss and time saved to compute the mask.

5.4. Feature Masking Tool

As we have seen in previous chapters exploring various methods to select optimal features, datasets tend to have a relatively high number of redundant features, increasing both the training and inference time, as well as increasing the space needed to store the model, both in runtime memory and on disk. In the case of Mice protein for example, the number of features can be further reduced from 50/77 to 20/77, without compromising the quality of the final model as in Figure 5.26.

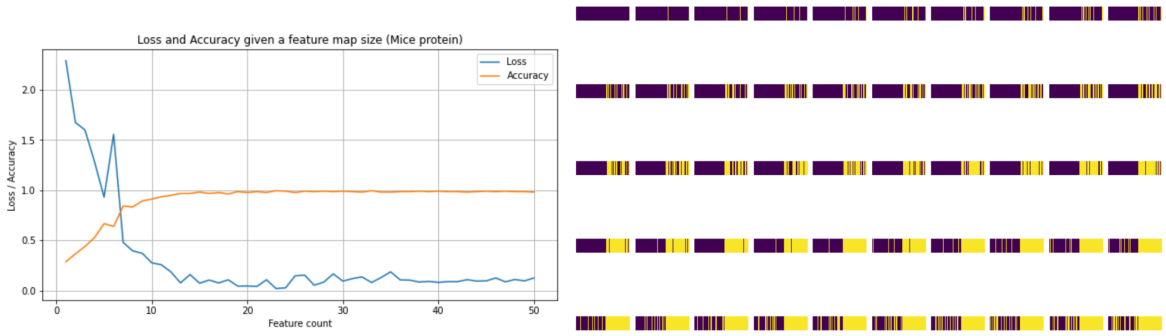

Figure 5.26: Performances of feature masking.

Building upon the results explored in this practical section of the thesis, we built an internal tool to get insights on how much the feature set can be reduced on any dataset, especially at large scale. The methodology is formalized in Algorithm 2. The goal is to help client teams understand the level of redundancy in their feature set, reduce it and speed up their machine learning flows, ideally at the same quality as the model trained with the full dataset.

5.4.1. Input

At a high level, the tool would need a dataset, a number of feature set sizes (“find best masks for 10%, 20%, … , 50% of the data”), and, optionally, a model to run experiments on. In case the model is absent, we will use a generic fully connected MLP to learn the optimal masks. The goal is to compute the best feature mask for each of the sizes in the budget and report on the quality of the model.

5.4.2. Output

For each size in the budget, the pipeline returns the following:

- Best mask: given the available sizes, this field contains the features that achieve the highest quality after training the model on them.

Algorithm 2 Feature Masking Algorithm

```
1:  function FeatureMasking(dataset, model, feature_set_sizes)
2:      feature_set_quality ← []
3:      best_feature_set ← None
4:      best_feature_set_quality ← None
5:      for i = 0 to |feature_set_sizes| do
6:          size ← feature_set_sizes[i]
7:          feature_set ← get_optimal_feature_set(dataset, model, size)
8:          quality ← model.evaluate(dataset.subset(feature_set))
9:          feature_set_quality ← feature_set_quality + quality
10:         if best_feature_set_quality < quality then
11:             best_feature_set_quality ← quality
12:             best_feature_set ← feature_set
13:         end if
14:     end for
15:     optimal_mask ← get_optimal_mask(feature_set_sizes, feature_set_quality)
16:     output ← (best_feature_set, optimal_mask, feature_set_quality)
17:     return output
18: end function
```

- Optimal mask: instead of returning the feature set that maximizes the quality, this field contains the mask that optimizes the tradeoff between the number of features and the quality of the mode.
- Mask qualities: the quality for the optimal feature subset for each of the sizes in the budget. The end user can use this to decide what is a reasonable tradeoff for their use case.

5.4.3. Architecture

The pipeline is modeled as a directed graph of jobs, each having a specified input, output and task. The main job receives the list with number of features and a dataset, then it creates K workers (K = | budget size |), each tasked to return the best mask with their Ki number of features. Once all the workers are done with their computation, another job is started, that collects their answers and computes the best and optimal masks. Once this part is done, the results are returned to the caller of the pipeline, as in Figure 5.27.

5.4.4. Usage

Having such a tool, internal clients can identify potential ways to make their machine learning flows more efficient, from both a training, inference and model storage perspective. The tool is meant to provide insights on how many features can be further reduced while keeping the same quality.

For the first iteration, the pipeline will run Sequential Attention to compute the masks, however, in the future, more complex approaches can be used, such as using intermediate Sequential Attention, or active learning with small subsets of the data, to speed up the pipeline and return results to users as quickly as possible.

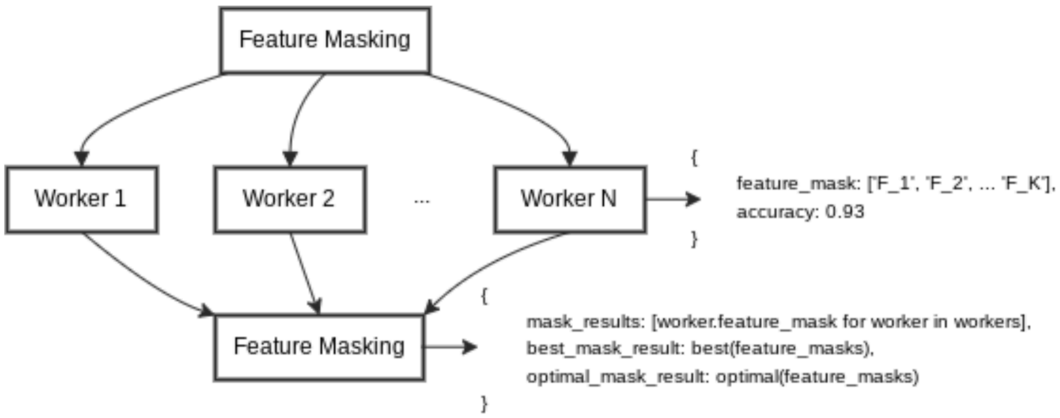

Figure 5.27: Feature masking tool.

5.5. Results and Impact

Looping back to the beginning of the thesis, the initial goal was to find ways to run machine learning experiments faster, reduce the time for a whole ML pipeline and reduce the feature set as much as possible.

With the Sequential Attention being the state of the art, we built on top of it to explore potential areas for improvement, such as ensembling, further feature set reduction and running in low data regimes.

Not only did we show that running simple Sequential Attention with less features is feasible and returns comparable results to the benchmark set by the paper, but we also explored other research directions, aimed to minimize the training time and provide insights on how little data is in fact needed to train a high quality ML model.

5.5.1. Training time saving

Having a good feature mask can dramatically reduce the training time. Since the number of features linearly impacts the time it takes to train a deep neural network, reducing the number of features by a non-trivial amount of X% would also reduce the training time by a similar percentage. This assumption holds true only if the architecture of the model is reduced to reflect the reduced size of the dataset. This has a direct implication on the model size as well.

For starters, one could zero-out the features not included in the reduced feature set, however this would create a very sparse model, with redundant connection. Rearchitecting the model to reflect the new reduced feature results in a lower model size, when saved after training and served for inference.

In Table 5.3, the tradeoff between quality and reduced model size is clear. Having a low but relevant number of parameters drops the accuracy with single digit points, while the size of the model drops with more than 90%, and so does the training and inference speed.

Table 5.3: Results

Model	Architecture	Params	Size KB)	Size relative to full model	Accuracy
Full-feature MLP on mnist	[784, 67, 10]	53275	208.11	100.0%	99.87%
50 feature MLP on mnist	[50, 32, 10]	1962	7.66	3.6%	95.11%
30 feature MLP on mnist	[30, 20, 10]	830	3.24	1.5%	93.58%

5.5.2. Reduced datasets

Having Sequential Attention or any of the aforementioned algorithms compute a mask that yields good performances might determine teams to deprecate features that do not have a direct contribution to training or inference tasks. Moreover, combined with the cost of feature acquisition, this approach might provide reductions in both storage and costs. Also, a future research direction could be looking into the explainability of models trained on reduced datasets. Having less redundancy might also lead to better visualizations.

5.5.3. Reduced inference time

Not only the training time is reduced having a smaller model and dataset, but also the inference time will drop, improving the user experience of the products that use those models. There might be a good tradeoff between a slightly expensive process of finding the mask and the reduction in time perceived by the end user.

5.5.4. New research directions

1. Parallelizing ensemble sequential attention.
2. Model explainability given reduced feature sets.
3. Active learning for using the least possible amount of data, while keeping the quality of the model high.
4. Finding an optimal tradeoff between the quality drop and the reduced time to train the full model.

5.5.5. Code

Given the surprising finding that no paper in the SLR had the code published to make it easier to replicate, we decided to publish the code behind this research. Together with the open source code published by Yasuda et al. [52], we think this will be a good replication package for follow-up work. The code can be found on GitHub (google-research /sequential_attention /ensembled_sequential_attention).

6
Conclusion

At the beginning of the thesis, we set the goal of making machine learning more efficient, by zooming into the data component of a machine learning system. Throughout this thesis, we went from understanding the current state of feature selection in the context of big infrastructures. Moreover, as previous studies look mostly into optimizing accuracy/loss for the test set, we look into more complex improvements, from the perspective of both minimizing a loss metric, but also lowering the time for training an optimal model, as well as lowering the memory footprint of a model training.

We started by looking into the background literature, finding some interesting studies that look into data redundancy, as well as into common problems when it comes to designing corporate scale data backed systems. The main problems with huge data volumes are Volume, Variety, and Velocity. With Volume, there is a challenge in training quickly models and finding redundancy in such an amount of data. With Variety, there is a challenge in training a model without bias, that is flexible enough to consider all the outliers. With Velocity, the main question that comes up is the speed one is able to train new models as the data distribution changes or keep existing models up to date.

The SLR offered a more in depth look over the research conducted in the last years on the topic of feature selection and data redundancy, in general. Apart from coming up with a list of reproducible steps for future research to use this as an anchor in the machine learning literature, some interesting insights came out of it. First and foremost, we saw that most of the studies we discovered were domain agnostic, the fact that helps advance the existing research. A domain agnostic approach can be applied in a multitude of scenarios, can be extended without losing its agnostic character, and can serve as the base for overly optimized domain specific applications. Speaking of domain specificity, some industries do prefer such approaches, since optimizing the accuracy of models is more important than finding trade-offs between accuracy and speed, for example. Such domains are healthcare and fintech, where, understandably the quality of the data and models they use can be crucial in impacting humans and conducting business.

From the solutions we surveyed, we've seen that most data and feature reduction approaches rely on heuristic methods, with filter feature selection algorithms being the most popular, due to their time efficiency and model interoperability. The main takeaway from the analyses we surveyed is that there is no free lunch when it comes to feature reduction, meaning that in order to optimize a metric there is a high chance of another metric being compromised, while they agree that in many business scenarios, having prior domain knowledge is a key factor in deciding what is data redundancy and what features are indeed valuable for training a highly predictive model. Along the

same lines, reviews that we included in our SLR conclude that a lack of focus in the business part of Machine Learning can impact decisions taken by both practitioners, as well as the ones made by the algorithms.

Having done the SLR, we shifted focus towards a more applied setting, where we aimed to change the way feature selection is done at a corporate scale. We partnered with Google to advance the state-of-the-art set by Google researchers in early 2023, through the Sequential Attention for Feature Selection paper. While Yasuda et al. [52] did advance the SOTA when it comes to model accuracy with reduced feature count, we considered other metrics as important in this optimization. Thus, we started looking into ways of finding a good equilibrium between model quality and the time it takes to reach it. We thus looked into ensemble models, that could collectively reach a consensus that is better in both time and quality than the previous state of the art. Seeing that redundancy problems start to show up, we came up with the Intermediate Sequential Attention algorithm, that would ensemble multiple workers with periodic syncs, so that the benefit of ensembling is still present, while redundancy is minimized through communication between models. We also showed that this flow achieves similar quality to the state of the art while minimizing the time it takes to reach it. To complement this vertical of research, we explored ways of coming up with an optimal feature subset size faster, looking into the usage of Singular Value Decomposition for this purpose. Lastly, we put all this work together in a scalable tool used internally by teams at Google, so that the research we conducted can start impacting products and consumers as soon as possible.

In essence, this thesis emphasizes the relationship between academic research and industry application, with the shared goal of improving machine learning systems, and working with data as efficiently as possible, without compromising quality. While academic studies lay the groundwork by providing insights, theories, and methodologies, it is the real-world application, as experienced at Google, that tests, refines, and applies them in a real-world setting the research findings. This iterative process of learning and applying is what propels the machine learning field forward, ensuring its relevance and impact in both the world of academia and the ever-evolving tech industry.

References

[1] Ashraf Abd El-Sattar, Nagy R Darwish, and Hesham Hefny. “A Survey Machine Learning Techniques on Big-Data Clustering”. In: *The 54 th Annual Conference on Statistics, Computer Sciences and Operations Research., Cairo, Egypt*. 2019, p. 131.

[2] Hervé Abdi and Lynne J Williams. “Principal component analysis”. In: *Wiley interdisciplinary reviews: computational statistics* 2.4 (2010), pp. 433–459.

[3] Akmal Akhatov and Shokh Abbos Ulugmurodov. “Training data selection and labeling for machine learning braille recognition models”. In: *International Journal of Contemporary Scientific and Technical Research* Special Issue (2023), pp. 15–21.

[4] Jose M Alvarez and Mathieu Salzmann. “Learning the Number of Neurons in Deep Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc., 2016. url: https://proceedings.neurips.cc/paper_files/paper/2016/file/6e7d2da6d3953058db75714ac400b584-Paper.pdf.

[5] B Amarnath, S Balamurugan, and Appavu Alias. “Review on feature selection techniques and its impact for effective data classification using UCI machine learning repository dataset”. In: *Journal of Engineering Science and Technology* 11.11 (2016), pp. 1639–1646.

[6] Mihai Anton. “Automated Machine Learning using Evolutionary Algorithms”. In: *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE. 2020, pp. 101–107.

[7] Sikha Bagui and Kunqi Li. “Resampling imbalanced data for network intrusion detection datasets”. In: *Journal of Big Data* 8.1 (2021), pp. 1–41.

[8] Azhari Shouni Barkah et al. “Impact of Data Balancing and Feature Selection on Machine Learning-based Network Intrusion Detection”. In: *JOIV: International Journal on Informatics Visualization* 7.1 (2023), pp. 241–248.

[9] Vighnesh Birodkar, Hossein Mobahi, and Samy Bengio. “Semantic Redundancies in Image-Classification Datasets: The 10% You Don’t Need”. In: *arXiv preprint arXiv:1901.11409* (2019).

[10] Andrei Z. Broder et al. “Search Advertising Using Web Relevance Feedback”. In: *Proceedings of the 17th ACM Conference on Information and Knowledge Management*. CIKM ’08. Napa Valley, California, USA: Association for Computing Machinery, 2008, pp. 1013–1022. isbn: 9781595939913. doi: 10.1145/1458082.1458217. url: https://doi.org/10.1145/1458082.1458217.

[11] Erik Brynjolfsson and Kristina McElheran. “The rapid adoption of data-driven decision-making”. In: *American Economic Review* 106.5 (2016), pp. 133–39.

[12] Girish Chandrashekar and Ferat Sahin. “A survey on feature selection methods”. In: *Computers & Electrical Engineering* 40.1 (2014), pp. 16–28.

[13] *Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says*. https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/?sh=587ce5a46f63. Accessed: 2023-01-18.

[14] Cody Coleman et al. “Selection via proxy: Efficient data selection for deep learning”. In: *arXiv preprint arXiv:1906.11829* (2019).

[15] Jennifer L Davidson and Jaikishan Jalan. “Feature selection for steganalysis using the Mahalanobis distance”. In: *Media forensics and security II*. Vol. 7541. SPIE. 2010, pp. 26–37.

[16] Xibin Dong et al. “A survey on ensemble learning”. In: *Frontiers of Computer Science* 14.2 (Apr. 2020), pp. 241–258. issn: 2095-2236. doi: 10.1007/s11704-019-8208-z. url: https://doi.org/10.1007/s11704-019-8208-z.

[17] *Feedback Loops in Machine Learning Systems*. https://towardsdatascience.com/feedback-loops-in-machine-learning-systems-701296c91787. Accessed: 2023-02-08.

[18] *Google Search Statistics*. https://www.internetlivestats.com/google-search-statistics/. Accessed: 2023-01-16.

[19] Venkat Gudivada, Amy Apon, and Junhua Ding. “Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations”. In: *International Journal on Advances in Software* 10.1 (2017), pp. 1–20.

[20] *How Much Data Is Created Every Day in 2023?* https://techjury.net/blog/how-much-data-is-created-every-day/. Accessed: 2023-01-11.

[21] *In-depth guide to machine learning in the enterprise*. https://www.techtarget.com/searchenterpriseai/In-depth-guide-to-machine-learning-in-the-enterprise. Accessed: 2023-01-18.

[22] Yuna Jeong, Myunggwon Hwang, and Wonkyung Sung. “Training data selection based on dataset distillation for rapid deployment in machine-learning workflows”. In: *Multimedia Tools and Applications* (2022), pp. 1–16.

[23] George H John, Ron Kohavi, and Karl Pfleger. “Irrelevant features and the subset selection problem”. In: *Machine learning proceedings 1994*. Elsevier, 1994, pp. 121–129.

[24] Petra J Jones et al. “Feature selection for unsupervised machine learning of accelerometer data physical activity clusters–A systematic review”. In: *Gait & Posture* 90 (2021), pp. 120–128.

[25] Utkarsh Mahadeo Khaire and R Dhanalakshmi. “Stability of feature selection algorithm: A review”. In: *Journal of King Saud University-Computer and Information Sciences* 34.4 (2022), pp. 1060–1073.

[26] Taghi Khoshgoftaar et al. “First order statistics based feature selection: A diverse and powerful family of feature seleciton techniques”. In: *2012 11th International Conference on Machine Learning and Applications*. Vol. 2. IEEE. 2012, pp. 151–157.

[27] Sun Hye Kim and Fani Boukouvala. “Machine learning-based surrogate modeling for data-driven optimization: a comparison of subset selection for regression techniques”. In: *Optimization Letters* 14.4 (2020), pp. 989–1010.

[28] Yiwen Liao, Raphaël Latty, and Bin Yang. “Feature Selection Using Batch-Wise Attenuation and Feature Mask Normalization”. In: *2021 International Joint Conference on Neural Networks (IJCNN)*. 2021, pp. 1–9. doi: 10.1109/IJCNN52387.2021.9533531.

[29] Huan Liu, Hiroshi Motoda, and Lei Yu. “A selective sampling approach to active feature selection”. In: *Artificial Intelligence* 159.1-2 (2004), pp. 49–74.

[30] Kumar N Neeraj and V Maurya. “A review on machine learning (feature selection, classification and clustering) approaches of big data mining in different area of research”. In: *Journal of Critical Reviews* 7.19 (2020), pp. 2610–2626.

[31] Alicia Parrish et al. “Does Putting a Linguist in the Loop Improve NLU Data Collection?” In: *arXiv preprint arXiv:2104.07179* (2021).

[32] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. “Deep learning on a data diet: Finding important examples early in training”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 20596–20607.

[33] Alec Radford et al. “Language models are unsupervised multitask learners”. In: *OpenAI blog* 1.8 (2019), p. 9.

[34] Murad Al-Rajab, Joan Lu, and Qiang Xu. “Examining applying high performance genetic data feature selection and classification algorithms for colon cancer diagnosis”. In: *Computer methods and programs in biomedicine* 146 (2017), pp. 11–24.

[35] Jashanpreet Singh Sadioura, Satbir Singh, and Amitava Das. “Selection of suboptimal feature set of network data to implement Machine Learning models to develop an efficient NIDS”. In: *2019 International Conference on Data Science and Engineering (ICDSE)*. IEEE. 2019, pp. 120–125.

[36] Arun Thundyill Saseendran et al. “Impact of data pruning on machine learning algorithm performance”. In: *arXiv preprint arXiv:1901.10539* (2019).

[37] D. Sculley et al. “Hidden Technical Debt in Machine Learning Systems”. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015. url: https://proceedings.neurips.cc/paper/2015/file/86df7dcfd896fcaf2674f757a2463eba-Paper.pdf.

[38] Luai Al-Shalabi. “New feature selection algorithm based on feature stability and correlation”. In: *IEEE Access* 10 (2022), pp. 4699–4713.

[39] Janvier Omar Sinayobye et al. “Hybrid model of correlation based filter feature selection and machine learning classifiers applied on smart meter data set”. In: *2019 IEEE/ACM Symposium on Software Engineering in Africa (SEiA)*. IEEE. 2019, pp. 1–10.

[40] Qinbao Song, Jingjie Ni, and Guangtao Wang. “A fast clustering-based feature subset selection algorithm for high-dimensional data”. In: *IEEE transactions on knowledge and data engineering* 25.1 (2011), pp. 1–14.

[41] Ben Sorscher et al. “Beyond neural scaling laws: beating power law scaling via data pruning”. In: *arXiv preprint arXiv:2206.14486* (2022).

[42] Patrick Stanula, Amina Ziegenbein, and Joachim Metternich. “Machine learning algorithms in production: A guideline for efficient data source selection”. In: *Procedia CIRP* 78 (2018), pp. 261–266.

[43] Ikbal Taleb et al. “Big data quality framework: a holistic approach to continuous quality management”. In: *Journal of Big Data* 8.1 (2021), pp. 1–41.

[44] Denny Thaler et al. “Training Data Selection for Machine Learning-Enhanced Monte Carlo Simulations in Structural Dynamics”. In: *Applied Sciences* 12.2 (2022), p. 581.

[45] Mariya Toneva et al. “An empirical study of example forgetting during deep neural network learning”. In: *arXiv preprint arXiv:1812.05159* (2018).

[46] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL].

[47] Benjamin SC Wade et al. “Machine learning on high dimensional shape data from subcortical brain surfaces: A comparison of feature selection and classification methods”. In: *Pattern Recognition* 63 (2017), pp. 731–739.

[48] Jue Wang, Kun Guo, and Shouyang Wang. “Rough set and Tabu search based feature selection for credit scoring”. In: *Procedia Computer Science* 1.1 (2010), pp. 2425–2432.

[49] Mei Wang, Xinrong Tao, and Fei Han. “A new method for redundancy analysis in feature selection”. In: *2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence*. 2020, pp. 1–5.

[50] Chathuranga Widanapathirana et al. “Automated Inference System for End-To-End Diagnosis of Network Performance Issues in Client-Terminal Devices”. In: *International Journal of Computer Networks Communications (IJCNC)* 4 (Apr. 2012), pp. 37–56.

[51] Jaekyung Yang and Sigurdur Olafsson. “Optimization-based feature selection with adaptive instance sampling”. In: *Computers & Operations Research* 33.11 (2006), pp. 3088–3106.

[52] Taisuke Yasuda et al. *Sequential Attention for Feature Selection*. 2023. arXiv: 2209.14881 [cs.LG].

[53] Fan Zhang et al. “Data driven feature selection for machine learning algorithms in computer vision”. In: *IEEE Internet of Things Journal* 5.6 (2018), pp. 4262–4272.

[54] Yong Zhang et al. “Feature selection algorithm based on bare bones particle swarm optimization”. In: *Neurocomputing* 148 (2015), pp. 150–157.

[55] Zheng Zhao and Huan Liu. “Searching for interacting features in subset selection”. In: *Intelligent Data Analysis* 13.2 (2009), pp. 207–228.