2022

A Hybrid Data and Hypothesis-Driven Model for Software Development in Support of the Machine-Learning Paradigm

Anthony D. Bowman
University Of Alabama At Birmingham

A HYBRID DATA AND HYPOTHESIS-DRIVEN MODEL FOR SOFTWARE DEVELOPMENT IN SUPPORT OF THE MACHINE-LEARNING PARADIGM

by

ANTHONY D. BOWMAN

LEON JOLOLIAN, COMMITTEE CHAIR
ARIE NAKHMANI
MAQBOOL PATEL
MURAT TANIK
EARL WELLS

A DISSERTATION

Submitted to the graduate faculty of The University of Alabama at Birmingham,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

BIRMINGHAM, ALABAMA

2022

A HYBRID DATA AND HYPOTHESIS-DRIVEN MODEL FOR SOFTWARE DEVELOPMENT IN SUPPORT OF THE MACHINE-LEARNING PARADIGM

ANTHONY DEAN BOWMAN

COMPUTER ENGINEERING

ABSTRACT

Historically, research has often been conducted in a hypothesis-driven manner with software development methodologies created to support those efforts. However, in recent years data-driven approaches to research have seen a dramatic rise in prominence. While software development methodologies such as agile development, extreme programming, and the waterfall model have allowed developers to tackle increasingly complex problems, they were not designed to efficiently support data-driven approaches such as the machine learning paradigm. To address the need to support the different programmatic requirements of both classical, hypothesis-driven as well as data-driven development, novel development strategies are warranted. In this research, we adapted the well-established spiral model to support both hypothesis- and data-driven development within its iterative design. Within this model, we included a novel framework for embracing the machine learning paradigm. By removing artificial limitations in the number and selection of machine learning algorithms and feature sets we have often seen in previous literature, this framework allows for the expanded application of machine learning techniques. Supported by parallelism, feature engineering, and the reuse of data and feature subsets, this framework supports the efficient exploration of both the problem and solution spaces. To demonstrate its benefits, we applied this updated lifecycle model to a complex neurological problem.

The results from this case study show this lifecycle model now provides greater flexibility for the developer in tailoring solutions to the ever-changing needs of a project, be they hypothesis- or data-driven. Our framework allows for greater adoption of the machine learning paradigm, providing support for developers to efficiently expand the scope of their work while generating more optimal results.

Keywords: software development lifecycle, hypothesis-driven research, data-driven research, machine learning, signal analysis, frequency analysis

DEDICATION

For my family.

ACKNOWLEDGMENTS

This work would not be possible without the continual guidance of my dissertation advisor and ever-welcome feedback from my committee members and fellow graduate students. I thank each of them for their support and understanding in improving and solidifying the scientific value of this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Overview

A variety of software development methodologies have been created over the past half century to support the creation of software that fulfills requirements of ever-increasing complexity. While these methodologies have allowed for the development of demanding software solutions, they were not designed to support the data-driven solutions which have recently become in high demand. The different programmatic requirements between the classical, algorithmic software development and data-driven development warrant the creation of a novel methodology with the flexibility to support both, allowing the developer to adapt to the needs of the project while utilizing the same development methodology. To accomplish this, we chose to adapt the spiral model, a well-established lifecycle model for software development with the iterative properties needed to incorporate our changes (Boehm, 1988). Within this adapted spiral model is our novel framework for the application of the machine learning paradigm (Bowman, Prabhakar, & Jololian, 2022). Through the reuse of data and feature sets, this framework allows for the efficient and rapid expansion of a project's scope by facilitating parallel, data-driven investigations. Through parallel model construction from a variety of machine learning algorithm and feature subsets, this framework generates a more optimal solution for each investigation. During the course of our research, we also realized our framework could be the basis for an introductory course in machine learning (Bowman &

Jololian, 2021). We applied our framework to a complex neurological problem, producing better results than previous literature (Bowman, Conwell, and Jololian, 2022). Expanding this application, we provide a detailed description of our adapted spiral model and a walkthrough of development within the context of the neurological case study.

Common Limitations in Current Machine Learning Applications

Advances in machine learning techniques have allowed researchers to make substantial gains in tackling difficult problems in diverse fields such as medical diagnosis, image recognition, and natural language processing. However, often times, machine learning has been used as a blunt instrument that is designed to ingest large amounts of raw data and produce usable predictive models that can help in decision making. Two bottlenecks inherent in this process are: (1) the ever increasing size of the data may slow down the development of solutions; this is primarily related to large number of features characterizing the dataset and computing hardware limitations, and (2) the way machine learning solutions are designed to answer a particular question as opposed to a set of questions related to the data; this is largely due to the nature of the scientific method and not an implicit limitation of the machine learning approach. Committing substantial resources to power this machine learning tool may still produce subpar results due to another dilemma inherent to machine learning: The combination of classification algorithm and feature set that will produce the best results cannot currently be predicted for any arbitrary data set or research question. In this way, machine learning research can benefit from both hardware and software optimizations to improve current and future analytic capacity in the search for the best possible results.

Currently, machine learning methodology is often employed as a general-purpose tool to process a large data set and produce a trained predictive model for a particular research question related to said data set. While this approach has produced valuable insight into both the respective subject domains and the machine learning techniques themselves, it involves a constant struggle of optimization which cannot simply be solved by acquiring faster hardware: Given a set of hardware, there exists at least one data set that exceeds its capacity, therefore the need for better, faster hardware is ever-present. Consequently, one must also turn to software level optimization to further improve their results beyond what is capable by their current hardware. In this way, software optimizations can allow more work to be accomplished within the hardware's current capacity. In addition, those optimizations will still play a role in making full use of any extra processing and/or storage capacity gained from newer hardware. Thus, software level optimizations through improving the overall process or algorithms can provide results that scale to both current and future hardware infrastructure. In this manner, we describe a framework with embedded software optimizations that can be scaled to an arbitrary size to fit current and future hardware limitations.

BACKGROUND

Artificial intelligence (AI) in its various forms has rapidly advanced in recent years to include applications in a wide variety of problem domains. We include here a book chapter we have submitted for publication by Elsevier which provides an overview of AI and examples of state-of-the-art applications.

Chapter 1 Introduction to Machine Learning and Artificial Intelligence Algorithms

ANTHONY D. BOWMAN, LEON JOLOLIAN

Submitted to Elsevier

Format adapted for dissertation

The machine learning paradigm and other data-driven approaches have seen a substantial rise in prominence in recent years. Fueled by advancements in data storage and wider availability of hardware with the processing power needed to make use of such algorithms, machine learning has allowed researchers to tackle increasingly complex problems in a wide variety of domains. In this chapter, the authors will provide a survey of different forms of artificial intelligence, including natural language processing and various machine learning methodologies.

This chapter begins with a brief introduction covering some of the history and computational advancements behind artificial intelligence (AI). The second section of this chapter provides further discussion of recent historical milestones in the surge of adoption seen just after the turn of the twenty-first century, followed by discussion on the difficulties still observed in developing standardized methods of applying AI techniques. The third section contains a survey of AI in its different forms with discussion on the variety of methodologies involved, including the different machine learning approaches such as supervised versus unsupervised, and predictive and generative algorithms. This will include examples showing the state-of-the-art capabilities currently in industry. During the discussion of supervised machine learning methodology, the authors provide further examples showing the variety and scope previous research has employed machine learning methodology within the context of diabetes research. The fourth section of this chapter provides discussion on some of the ethical concerns related to AI and its applications. In this chapter's fifth section the authors will then discuss the inclusion of the machine learning paradigm within a model of research that employs both data- and hypothesis-driven approaches. Following this model enables researchers to benefit from the respective advantages of each approach as they complement each other, thereby providing the researchers with a more complete model for their scientific endeavors.

1.1 Introduction

In the latter half of the twentieth century, algorithms were developed that could generate predictive models based on the systematic processing of supplied data. Because of their potential to mirror natural learning from experience, these algorithms were later labeled as machine learning algorithms. These algorithms were then incorporated into a larger corpus of machine learning methodology, a type of data-driven research. While the problem-solving potential of these algorithms was recognized in theory, it could not begin to be realized until the turn of the twenty-first century when technological advances in parallel processing and data storage reached a critical stage. At this point, improvements in manufacturing allowed for increased data density at lower prices than ever before, opening up the possibility of storing the ever-increasing amounts of data being generated by businesses and institutions as society gradually embraced the digital age.

The incorporation of intelligent systems and machine learning algorithms into a wide variety of application domains arguably began gaining significant momentum in the late 1990s before reaching critical mass in the early 2000s. During this period, intelligent systems stirred interest in the field after the system dubbed Deep Blue by IBM defeated the human chess champion Garry Kasparov (Campbell, Hoane Jr, & Hsu, 2002; Newborn, 2012). Since then, similar chess programs have been developed which also are capable of routinely defeating human grand champions. They typically accomplish this by generating an n-ary decision tree with every branch representing a possible, legal chess move on the board. Each move is assigned a calculated weight representing how favorable that sub-tree is to the program toward victory so that the program can then make the best decision possible given the current state of the game board. This approach at its core can then be considered a search problem i.e. searching through the generated possibilities in order to find the best possible path forward. The task of a machine learning algorithm may also be described this way: Given a training data set of features, search through all possible classification parameters to find the parameter set which allows for the best performance, provided time constraints. More specifically, a linear regression algorithm attempts to search for the values b and m which specify the linear function that best predicts the target value of the training data and subsequent test data (Seber & Lee, 2012). Similarly, a logistic regression algorithm attempts to search for the parameters that describe a non-linear equation which best classifies each data point in the training and test data sets (Bates & Watts, 1988). An artificial neural network greatly expands on this task by searching for the best set of weights between every layer, affording it greater power to identify and incorporate more subtle details present in the feature set of the data at the cost of increased time requirements (Bates & Watts, 1988; Negnevitsky, 1995).

1.2 The Birth and Rise of Machine Learning

Many of the algorithms now commonly labeled as machine learning algorithms were first formalized decades ago, such as the perceptron, an early version of the artificial neural network (Rosenblatt, 1958). While these algorithms were published, there remained a lack of utilization due to limitations of computer hardware. This bottleneck of application began to relax in the late 2000s as two key aspects of hardware reached critical turning points: A dramatic increase in processing power driven in part by parallelization, and the availability of cheap, large capacity storage to contain the ever-growing collection of data to be processed. In order to make use of newly available processing power, large amounts of data must be collected and organized, and in the new digital age there was no shortage of data. As machine learning became more commonplace, engineers discovered another avenue of hardware to take advantage of: Graphical processing units (GPUs). The GPUs optimizations in calculating and displaying graphical data also translated well to the task of training machine learning models, particularly those that could be represented in hardware by a two-dimensional matrix such as the artificial neural network. More recent generations of GPUs contain specialized sub-units specifically for accelerating tasks related to machine learning as demand for such power has only continued to grow (Markidis et al, 2018).

Machine learning methodology has been used with great success to improve lives at the level of the community, the individual, and individual tissues. Such applications include the automatic screening of mammograms for potentially cancerous growths, the classification of epilepsy types, and the identification of individuals at elevated risk to develop diabetes. Machine learning has also been shown to be effective in creating models for use at the tissue level. Models for the classification of cell types using the Breast Cancer Wisconsin data set have shown to have over 99% accuracy when distinguishing between benign and malignant samples (Abdar & Makarenkov, 2019). A common theme seen in these examples is the detection of an anomaly among a normal background, be it a cancerous tumor, seizure activity, or a combination of factors leading to development of insulin resistance. Machine learning as a means of anomaly detection has also been applied in the financial sector to identify potentially fraudulent transactions. Through each of these examples, machine learning methodology has shown its potential to improve the quality of life of a community and its members.

Development of Artificial Intelligence and Machine Learning Methodologies

While other engineering disciplines have developed scientifically sound, robust methods to produce solutions through processes that have withstood rigorous testing over time, the field of artificial intelligence, including machine learning, is still by comparison in its early years. Because of this, such a scientific method to craft reliable solutions has not been developed, forcing developers and researchers to resort to ad hoc, trial and error methods. While wholly inefficient compared to tried-and-true processes, some of these efforts have borne fruit which have shown to produce acceptable results. In some cases, further development from this starting has led to surprising advances in our understanding of the machine learning paradigm and what we may accomplish with it.

1.3 Branches of Artificial Intelligence

Artificial intelligence (AI) is now an umbrella term that encompasses a variety of fields including but not limited to expert systems, natural language processing, and machine learning. While these fields have a wide variety of both applications and low-level methodologies, they share a common theme: The ability to learn from given data, use a model to perform a task in a manner similar to a human, then adjust that model based on feedback, thereby learning from the experience. This often involves the automatic processing of large amounts of data, constructing a mathematical model based on the data and a specific task, taking action in response to input, gathering feedback and refining the model. In this way, AI is intended to iteratively learn from experience in a similar manner as a human does, thus gradually becoming more intelligent.

Natural Language Processing

One branch of AI involving the processing and analysis of natural language is known as natural language processing, or NLP. This branch combines the domain knowledge of linguistics with the computational thinking of computer science in order to enable a computer to process input given in a natural language and respond accordingly. This is generally accomplished by analyzing and parsing the input into tokens according to syntactical structure, analyzing the semantics of each token, incorporating that analysis into the given context. Given input may be textual or spoken, providing additional technical and linguistic challenges according to the format of the input data. For spoken data, additional consideration must be given to account for such variables as accents, disfluency, verbal interruptions, and the quality of the audio recordings. While textual data is not affected by accents, researchers must still contend with the possibility of spelling or grammatical errors, as well as tokens from outside a given lexicon such as emojis.

Recommender Systems

A software tool, component, or technique that provides suggested related items believed to be of interest to the user is labeled a recommender system (Aggarwal, 2016;

Ricci et al, 2011). Sometimes called collaborative filtering models, the label was expanded after other techniques were developed that did not explicitly rely on filtering out negative alternatives, instead searching for and finding positive alternatives such as content-based modeling (Aggarwal, 2016). The topic of recommender systems has seen a dramatic surge in prominence in conjunction with the continued rise of e-commerce (Ricci et al, 2011). The growing need for businesses to be able to serve customers with recommendations on products for purchase is a prime example of previously small-scale systems being adapted to serve as a large-scale solution (Resnick & Varian, 1997). Systems previously created to serving book recommendations to users now became critical components in offering product recommendations across multiple product domains and vast databases (Aggarwal, 2016; Linden et al, 2003). Further refinement of these recommender systems created personalized offerings, allowing for more targeted recommendations thought to improve the odds of a sale. Such systems have also been deployed in industry for recommendation of items for content consumption, such as related web pages or similar movies or TV series on such commercial platforms as Netflix.

Machine Learning Algorithms

Machine learning algorithms process given input data in the search for patterns to build a mathematical model of that data. These models can be constructed in order to predict specific values or classify/categorize future input. In the case of the former, the model is an equation with parameters refined through training using a set of example data, resulting in the model receiving feature data and calculating the corresponding value. The user can then provide feature data for an unknown sample and have the model calculate the value it predicts that sample to have. An example of this would be if a possible were trained to predict the sale price of a home in a given zip code based on the number of bedrooms, bathrooms, square footage and size of the garage. The user could then feed the model with an example specifying those parameters and receive the estimated sale price based on the historical data used to create the model. If the user was only interested in distinguishing between homes that sold or did not sell, then a similar methodology could be used to create a model that classifies the data instead of predicting a specific value. In this application, the classification algorithm would be trained on data describing a variety of homes, with each example labeled as "sold" or "not sold." After pre-processing, the constructed model would then be able to receive the specified features of an unknown sample and predict whether that home will sell or not, based on the given historical data. This data-driven methodology then places a heavy emphasis on the quality and validity of the data as that has an immense role in the usefulness of the model created.

Table 1: Common machine learning algorithms and their characteristics

Algorithm	Type of Approach	Type of Model
Linear Regression	Supervised	Predictive
Logistic Regression	Supervised	Discriminative
BayesNet	Supervised	Discriminative
NaiveBayes	Supervised, Unsupervised	Discriminative, Generative
Artificial Neural Network	Supervised, Unsupervised	Discriminative, Generative
Decision Tree	Supervised	Discriminative

K-means Clustering	Unsupervised	Generative
Restricted Boltzmann Machine	Unsupervised	Generative

Unsupervised Learning

In one approach to machine learning, the algorithm is provided a set of examples as input with the task of organizing the input according to any patterns it finds in the features of those examples. In this approach each data point in the input is unlabeled, so the resulting model has no performance metric it can be judged by and, therefore, no error to be calculated. The algorithm is left up to its devices to search through the provided input, identify predominant patterns in the features of each example and organize those examples accordingly. The researcher is then able to perform their own analysis on the output model to discover what pattern(s) were found and any potential higher-level patterns were uncovered. As an example, a k-means clustering algorithm can be configured by the user for a given number of clusters and be provided examples of animals with features such as whether the animal has fur, gives birth to live offspring, or has a vertebra. The output model could then have the given number of clusters where one cluster is composed of examples that have fur, do give birth to live offspring, and have a vertebra. It would then be up to the user to identify this pattern and recognize that this cluster represents the mammals within the data set.

Supervised Learning

In another approach to machine learning, the algorithm is provided input data where each example has been associated with a label or tag. The task of the algorithm is then to search through the defined features of the input to identify patterns which correlate to the different labels, thereby associating those patterns to a classification system. This is done by building a model with the training data set, then using that model to classify examples from a test data set and calculating an error metric. The model is then adjusted based on that error and given another set of test data to classify. This cycle repeats until either the error is calculated to be below a certain threshold, or a time limit is reached. Testing a model and calculating the error for a given pass can be done by either using a subset of the training data or by holding some data separate from the process that the model has not seen before. The model is thereby a mapping of features to labels, describing some pattern in the features which characterize each label in the provided data. Previous research employing this approach include the aforementioned discrimination task between benign and malignant breast cancer cells (Abdar & Makarenkov, 2019). In keeping with this medical context, this approach has also been utilized in attempts to identify patients with type 2 diabetes, a disease increasingly common in younger age groups (Krishnamoorthi et al, 2022; Ragab et al, 2022; Sharma & Shah, 2021). In this case, models were trained based on factors believed to be correlated with one's risk for type 2 diabetes, including BMI, age, and, in the case of Ragab et al, processed images of patients' retina showing microvascular changes. In other literature, researchers explored a supervised machine learning approach providing the algorithm with health record data (Ganie et al, 2022; Haq et al, 2020). The resulting model was then capable of receiving features extracted from the electronic medical record and identify patients at high risk for having type 2 diabetes. Research reviewing the literature in this problem domain note overall success in applying machine learning techniques to this particular problem, although there is some disagreement in which algorithm should be the one of choice for future researchers (Fregoso-Aparicio et al, 2021; Sharma & Shah, 2021). This variance harkens back to the lack of a standardized methodology for applying machine learning techniques to a particular problem domain, resulting in the variability in results seen when building models with different algorithms and feature sets.

Generative Networks

A different approach to machine learning involves the construction of a model which can then produce additional data instead of individual values or classifications. Instead of the predictive or discriminative task, the generative model provides the user a data point that is similar to the data points in the training data used in its construction. It creates this new instance based on the characteristics of the data used to train it. Example algorithms that can be used in this approach include the naïve bayes, artificial neural network, the Restricted Boltzmann Machine and its variants (Salakhutdinov & Larochelle, 2010). Through this methodology researchers have been able to generate a variety of products such as programming code, natural language text, audio and images (OpenAI, May 2022; Ouyang et al, 2022). The architecture known as Codex has been created to generate executable programming code based on natural language input from a human user. While such an intellectual activity as programming is an impressive feat and considered by some to be a major milestone, such efforts typically necessitate the processing of enormous amounts of data and fine-tuning billions of parameters to produce acceptable results (Ouyang et al, 2022). To facilitate some of these efforts, organized databases of images have been created such as ImageNet (Deng et al, 2009). Creation of synthetic images has progressed to include synthetic animation of moving images, sometimes called deepfakes (Kim et al, 2018). Of course, this creates ethical dilemmas as the creation of such convincing synthetic data can have dramatic, real-world consequences if produced with ill intentions. Because of this, the developers of some models have introduced constraints on certain aspects such as filtering out generated images of real people (OpenAI, April 2022). While great strides have been made with these generative models, researchers have continued to seek automated methods of fine-tuning these models, which have led to the development of generative adversarial networks.

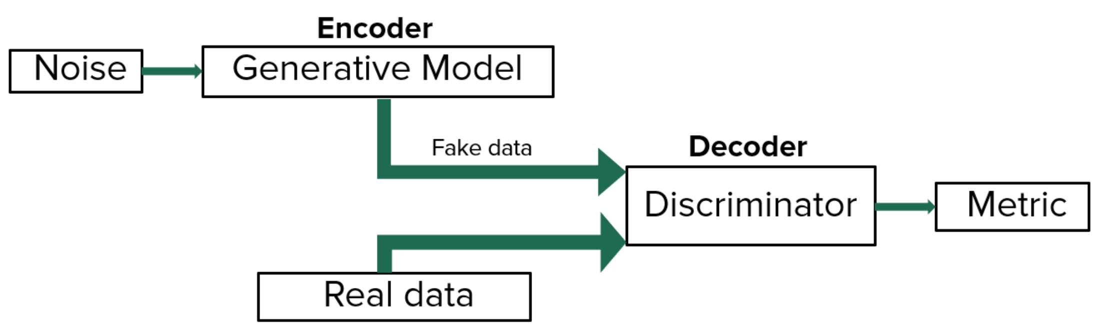


Figure 1: General framework of a Generative Adversarial Network

Adversarial Networks

The generative adversarial network (GAN) is the solution to the problem of automatic, large-scale feedback for fine-tuning generative models (Goodfellow et al, 2014). This approach involves two models: The generative model G and discriminative model D. Both models are initially trained in their respective task using real data for D and noise for G. They then engage in a zero-sum game where G generates an example and D attempts to determine if that example is from G (synthetic or "fake") or from the data set it was trained on ("real"). If D correctly classifies the example as being "fake," the parameters of G are adjusted, and another example is generated. This loop is repeated until D can only correctly identify the synthetic data less than half the time (or another threshold chosen by the user). This and similar approaches have been employed with great success in game theory, creating models which can now consistently win against skilled human opponents in games such as chess and Go ("*AlphaGo*," n.d.). Whereas previous efforts using tree-based searching and neural networks had proven successful, generative models in an adversarial framework have shown greater capabilities to generalize instead of becoming an expert at a single game ("*Alphazero: Shedding new light on chess, Shogi, and go,"* n.d.). In the case of Alphazero, the general adversarial approach can also be considered an example of reinforcement learning i.e. the model continually learns through trial and error, playing games against another model and adjusting internal parameters if it loses (negative reinforcement). With the reinforcement learning approach, models are trained to maximize a calculated reward within the confines of a dynamic environment (Kaelbling et al, 1996; Sutton & Barto, 2018). As opposed to the minimizing of a calculated error value in the supervised approach, reinforcement learning does not need labeled input, leaving it the capability to generalize and be applied to various domains, including prominently game theory (François-Lavet et al, 2018).

Time Series and Signal Processing

Included in the multitude of subject domains machine learning methodology has been applied to are problems which require the processing and analysis of data gathered in the form of a time series. The processing of signals can also be included in this problem domain as analysis can be performed in the time domain or frequency domain. This greatly expands the number of possible features researchers can consider for processing using machine learning algorithms to not only include features directly extracted from the original signal data, but also features extracted from transformed data e.g. frequency domain data. Compounding this is the possibility to compute additional time and frequency domain features used as metrics to quantitatively describe the relationship between the signals generated from different sources in the data set. For example, researchers could include power spectral features from a variety of frequency bands in addition to spectral coherence in the same or different frequency bands to compose large feature sets. An example subject domain that often encounters this scenario is the analysis of electroencephalogram (EEG), magnetoencephalogram (MEG), or electrocardiogram (ECG or EKG) data for signs of pathology.

Recent developments in applying machine learning methodology to processing medical signal data include the detection of seizure activity in EEG or MEG data as well as abnormal cardiac activity on the ECG. Wearable technology has also advanced to allow for machine learning to be incorporated into patients' lives in an unobtrusive manner. In the last few years, some studies have shown the Apple Watch to be effective in monitoring for cardiac arrythmia such as atrial fibrillation (Seshadri et al, 2020; Strik et al, 2020). This deployment of machine learning into small, wearable technology opens up new opportunities for monitoring and intervention in case early signs of a medical emergency are detected. In addition to this real-time monitoring, machine learning methodology has been applied to the problem of seizure localization. In this task, researchers have attempted to construct models to determine which region of the brain an epileptic seizure is originating from based on a patient's EEG or MEG data (Aoe et al, 2019; Guo et al, 2018; Soriano et al, 2017).

1.4 Ethical Concerns of AI

Ethical concerns about the creation and use of AI span as wide of a range as the applications of AI. While such a broad range is to be expected of any technology that touches so many different subject domains, AI in particular has historically caused great concern to some. These concerns have ranged from the pragmatic, grounded ideas related to privacy and truth (related to AI used in surveillance and synthetic image/audio creation) to the more spectacular ideas related to AI-driven machines rebelling against humanity. Focusing this discussion on the former, there have been privacy concerns raised in recent years regarding the increasing use of security cameras by governments and the machine learning software used to process the video data. Major concerns relate to the ability for government entities to track individuals throughout their day, depending on how widespread the surveillance system is along common routes. These concerns are extended to include the ability for wearable technology to track an individual's location even as they leave the view of any camera-based surveillance, providing corporations additional data about their customers.

1.5 Scientific Advancement

Two general approaches to research can be described as hypothesis-driven versus data-driven (Hulsen et al, 2019). Until recently, a significant portion of research was conducted using a hypothesis-driven approach where the researchers formulate a narrowly defined hypothesis, then design an experiment to test that hypothesis as specifically and precisely as possible, given current tools and methodology. This focused application of the scientific method allows for rigorous testing and validation of the hypothesis, ultimately aiming to produce knowledge that has withstood multiple passes of validation with reproducible results. In this way, hypothesis-driven research yields firmly grounded knowledge, but within a narrowly defined scope which limits the speed at which knowledge can be generated.

On the other hand, data-driven approaches to research allow for more rapid knowledge discovery through the processing of large volumes of data (Das et al, 2015; Hulsen et al, 2019). With the recent development of standardized tools, systematic data mining has become possible with ever-increasing volumes of data (Das et al, 2015). While this approach to building a model has generated some excitement in various research communities for its potentially unforeseen discoveries, some studies have discussed significant challenges (Hulsen et al, 2019). These include the philosophical underpinning behind any model produced by data-driven methods and applied to a subject domain without a thorough understanding and validation of what that model signifies (Callebaut, W., 2012). From a more practical standpoint, other studies have pointed out challenges in the reproducibility and generality of data-driven models with some finding no consensus between models (Mestre et al, 2018).

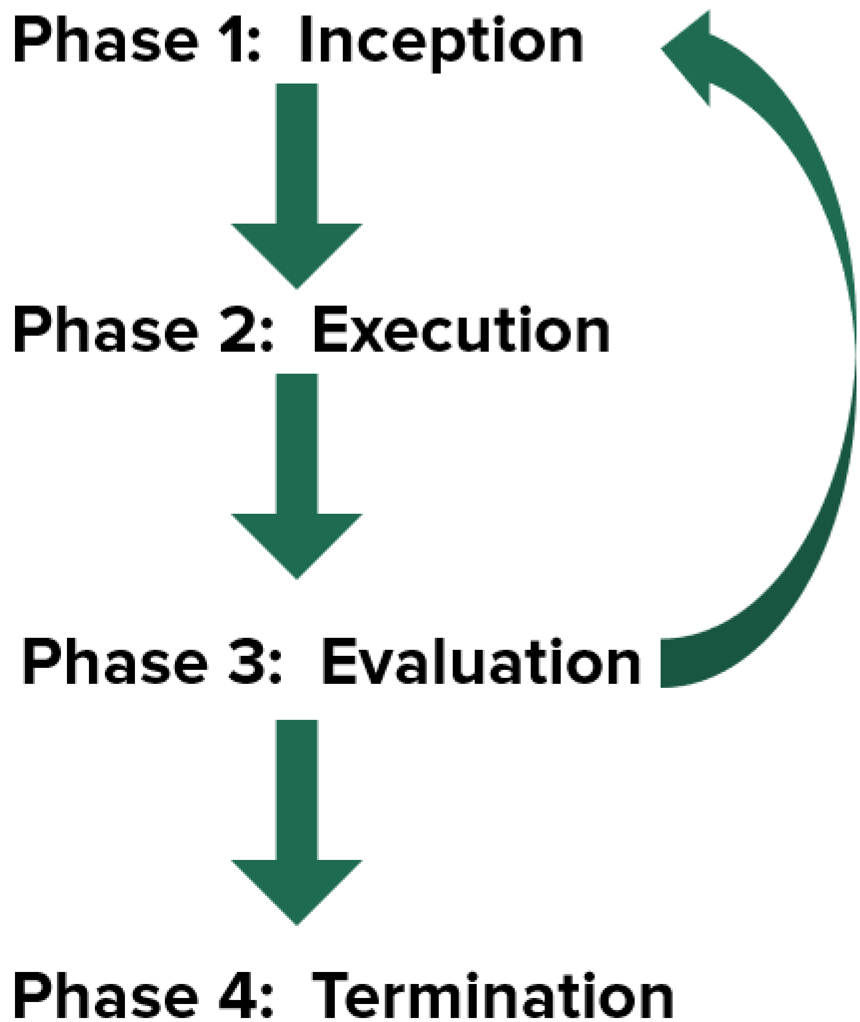


Figure 2: Simplified Lifecycle Model for Research

Recently, the authors have endeavored to include both approaches into a single model for conducting research resulting in the creation of the iterative model seen in Figure 1. This model incorporates a branching decision in Phase 1 that calls for the researcher(s) to consider which approach, data- or hypothesis-driven, they wish to adopt for the current iteration prior to executing an experiment which follows that approach in Phase 2. During their evaluation of results in Phase 3, they will review their results and determine if the goals of this research project have been met. If not, the project returns to Phase 1 where they may choose to adopt a different approach for the next iteration.

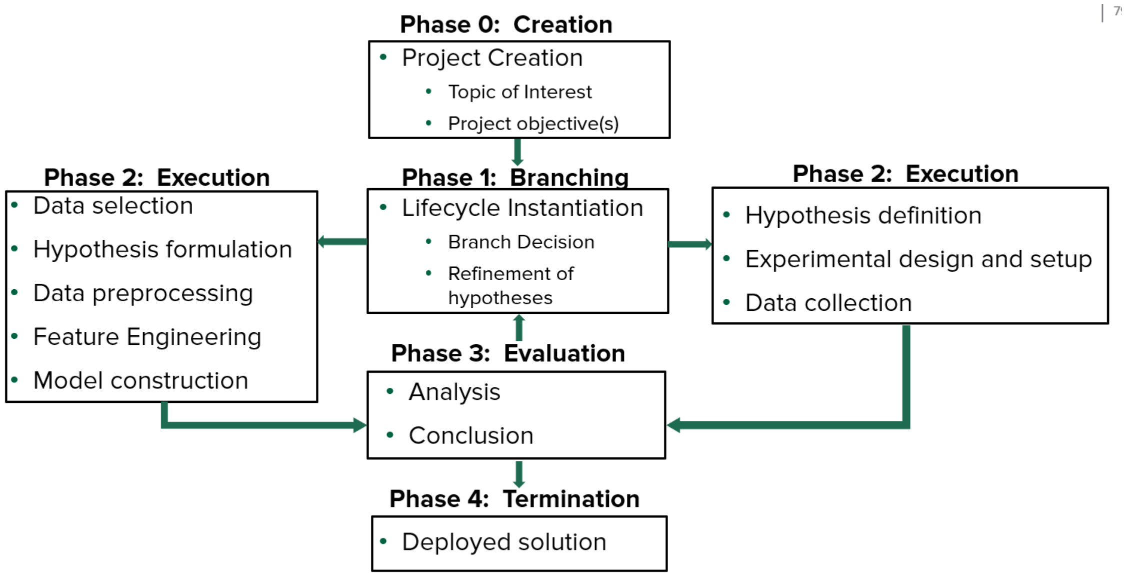


Figure 3: Detailed Lifecycle of Research

Figure 2 shows a more detailed and expanded model with high level processes in each phase. In this view the distinction between the data-driven and hypothesis-driven tracks on the left and right respectively is made clear, emphasizing their respective focus on handling data or generating valid data through a specific experimental design. Following execution of the chosen research track, the researcher(s) analyze their results and evaluate whether the goal(s) of the project has been achieved. If so, work proceeds to Phase 4 where project artifacts are delivered, and work is concluded. If the project's goals have not been met, work loops back to Phase 1. The working hypotheses may be refined by incorporating knowledge gained from the previous cycle. The researcher(s) may then decide on adopting a different approach for this current iteration, depending on how their needs have changed. In this way, they can benefit from the strengths of both data-driven and hypothesis-driven efforts in turn, reinforcing the validity of the knowledge generated through this model and strengthening the value of their efforts.

1.6 Conclusion

In this chapter the authors discussed the rise of the machine learning paradigm and rapid, widespread prevalence of artificial intelligence in a broad range of problem domains. Among these discussions was an assortment of AI categories, including different types of machine learning methodologies. Examples showing the state-of-the-art of these applications point to greater involvement of AI in all aspects of industrialized, modern life, from wearable technologies with real-time monitoring for medical emergencies to intelligent systems which automatically adjust for the changing needs of a city's population. A brief discussion covering ethical concerns related to some of these applications was included here. Finally, the authors presented a model of research which incorporates both hypothesis- and data-driven approaches to allow the researcher to benefit from the advantages of both.

References

Abdar, M., & Makarenkov, V. (2019). CWV-BANN-SVM ensemble learning classifier for an accurate diagnosis of breast cancer. *Measurement*, *146*, 557-570. https://doi.org/10.1016/j.measurement.2019.05.022

AlphaGo. DeepMind. (n.d.). Retrieved August 14, 2022, from https://www.deepmind.com/research/highlighted-research/alphago

Alphazero: Shedding new light on chess, Shogi, and go. RSS. (n.d.). Retrieved August 14, 2022, from https://www.deepmind.com/blog/alphazero-shedding-new-light-on-chess-shogi-and-go

Aoe, J., Fukuma, R., Yanagisawa, T., Harada, T., Tanaka, M., Kobayashi, M., ... & Kishima, H. (2019). Automatic diagnosis of neurological diseases using MEG signals with a deep neural network. *Scientific reports*, *9*(1), 1-9. https://doi.org/10.1038/s41598-019-41500-x

Bates, D. M., & Watts, D. G. (1988). *Nonlinear regression analysis and its applications*. Wiley. https://doi.org/10.2307/2289810

Callebaut, W. (2012). Scientific perspectivism: A philosopher of science's response to the challenge of big data biology. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*, *43*(1), 69-80. https://doi.org/10.1016/j.shpsc.2011.10.007

Campbell, M., Hoane Jr, A. J., & Hsu, F. H. (2002). Deep blue. *Artificial intelligence*, *134*(1-2), 57-83. https://doi.org/10.1016/S0004-3702(01)00129-1

Das, M., Cui, R., Campbell, D. R., Agrawal, G., & Ramnath, R. (2015, October). Towards methods for systematic research on big data. In *2015 IEEE International Conference on Big Data (Big Data)* (pp. 2072-2081). IEEE. https://doi.org/10.1109/BigData.2015.7363989

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255). IEEE. https://doi.org/10.1109/CVPR.2009.5206848

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*. https://doi.org/10.48550/arXiv.1406.2661

Guo, J., Yang, K., Liu, H., Yin, C., Xiang, J., Li, H., ... & Gao, Y. (2018). A stacked sparse autoencoder-based detector for automatic identification of neuromagnetic high frequency oscillations in epilepsy. *IEEE transactions on medical imaging*, *37*(11), 2474-2482. https://doi.org/10.1109/TMI.2018.2836965

Hulsen, T., Jamuar, S. S., Moody, A., Karnes, J. H., Orsolya, V., Hedensted, S., ... & McKinney, E. (2019). From big data to precision medicine. *Frontiers in medicine*, *6*, 34. https://doi.org/10.3389/fmed.2019.00034

Kim, H., Garrido, P., Tewari, A., Xu, W., Thies, J., Niessner, M., ... & Theobalt, C. (2018). Deep video portraits. *ACM Transactions on Graphics (TOG)*, *37*(4), 1-14. https://doi.org/10.1145/3197517.3201283

Markidis, S., Der Chien, S. W., Laure, E., Peng, I. B., & Vetter, J. S. (2018, May). Nvidia tensor core programmability, performance & precision. In *2018 IEEE international* *parallel and distributed processing symposium workshops (IPDPSW)* (pp. 522-531). IEEE. https://doi.org/10.1109/IPDPSW.2018.00091

Mestre, T. A., Eberly, S., Tanner, C., Grimes, D., Lang, A. E., Oakes, D., & Marras, C. (2018). Reproducibility of data-driven Parkinson's disease subtypes for clinical research. *Parkinsonism & related disorders*, *56*, 102-106. https://doi.org/10.1016/j.parkreldis.2018.07.009

Negnevitsky, M. (2005). *Artificial Intelligence: A Guide to Intelligent Systems. Second ed*. Retrieved from www.academia.dk. ISBN-13 978-0321204660

Newborn, M. (2012). *Kasparov versus Deep Blue: Computer chess comes of age*. Springer Science & Business Media. https://doi.org/10.1007/978-1-4612-2260-6

OpenAI. (2022, April 14). *Dall·E 2*. OpenAI. Retrieved August 14, 2022, from https://openai.com/dall-e-2/

OpenAI. (2022, May 23). *GPT-3 powers the next generation of apps*. OpenAI. Retrieved August 14, 2022, from https://openai.com/blog/gpt-3-apps/

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*. https://doi.org/10.48550/arXiv.2203.02155

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386. https://doi.org/10.1037/h0042519

Salakhutdinov, R., & Larochelle, H. (2010, March). Efficient learning of deep Boltzmann machines. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 693-700). JMLR Workshop and Conference Proceedings. https://doi.org/10.1162/NECO_a_00311

Seber, G. A., & Lee, A. J. (2012). *Linear regression analysis* (Vol. 329). John Wiley & Sons.

Seshadri, D. R., Bittel, B., Browsky, D., Houghtaling, P., Drummond, C. K., Desai, M. Y., & Gillinov, A. M. (2020). Accuracy of Apple Watch for detection of atrial fibrillation. *Circulation*, *141*(8), 702-703. https://doi.org/10.1002/9780471722199

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *nature*, *529*(7587), 484-489. https://doi.org/10.1038/nature16961

Soriano, M. C., Niso, G., Clements, J., Ortín, S., Carrasco, S., Gudín, M., & Pereda, E. (2017). Automated detection of epileptic biomarkers in resting-state interictal MEG data. *Frontiers in neuroinformatics*, *11*, 43. https://doi.org/10.3389/fninf.2017.00043

Strik, M., Caillol, T., Ramirez, F. D., Abu-Alrub, S., Marchand, H., Welte, N., ... & Bordachar, P. (2020). Validating QT-interval measurement using the Apple Watch ECG to enable remote monitoring during the COVID-19 pandemic. *Circulation*, *142*(4), 416-418. https://doi.org/10.1161/CIRCULATIONAHA.120.048253

Ganie, S. M., Malik, M. B., & Arif, T. (2022). Performance analysis and prediction of type 2 diabetes mellitus based on lifestyle data using machine learning approaches. *Journal of Diabetes & Metabolic Disorders*, 1-14. https://doi.org/10.1007/s40200-022-00981-w

Sharma, T., & Shah, M. (2021). A comprehensive review of machine learning techniques on diabetes detection. *Visual Computing for Industry, Biomedicine, and Art*, *4*(1), 1-16. https://doi.org/10.1186/s42492-021-00097-7

Fregoso-Aparicio, L., Noguez, J., Montesinos, L., & García-García, J. A. (2021). Machine learning and deep learning predictive models for type 2 diabetes: a systematic review. *Diabetology & Metabolic Syndrome*, *13*(1), 1-22. https://doi.org/10.1186/s13098-021-00767-9

Krishnamoorthi, R., Joshi, S., Almarzouki, H. Z., Shukla, P. K., Rizwan, A., Kalpana, C., & Tiwari, B. (2022). A novel diabetes healthcare disease prediction framework using machine learning techniques. *Journal of Healthcare Engineering*, *2022*. https://doi.org/10.1155/2022/1684017

Ragab, M., AL-Ghamdi, A. S., Fakieh, B., Choudhry, H., Mansour, R. F., & Koundal, D. (2022). Prediction of Diabetes through Retinal Images Using Deep Neural Network. *Computational Intelligence and Neuroscience*, *2022*. https://doi.org/10.1155/2022/7887908

Haq, A. U., Li, J. P., Khan, J., Memon, M. H., Nazir, S., Ahmad, S., Khan, G. A., & Ali, A. (2020). Intelligent machine learning approach for effective recognition of diabetes in E-healthcare using clinical data. *Sensors*, *20*(9), 2649. https://doi.org/10.3390/s20092649

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, *4*, 237-285. https://doi.org/10.1613/jair.301

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, *11*(3-4), 219-354. http://dx.doi.org/10.1561/2200000071

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press. ISBN-13 978-0262193986

Resnick, P., & Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, *40*(3), 56-58. https://doi.org/10.1145/245108.245121

Linden, G., Smith, B., & York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, *7*(1), 76-80. https://doi.org/10.1109/MIC.2003.1167344

Aggarwal, C. C. (2016). *Recommender systems* (Vol. 1). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-29659-3

Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender systems handbook* (pp. 1-35). Springer, Boston, MA. https://doi.org/10.1007/978-1-4899-7637-6_1

Limitations in Current Methodology

In machine learning, researchers often face a two-fold dilemma inherent to the field: First, current limitations of mathematically valid methods of predicting which combination of machine learning algorithm and feature set will yield optimal results in the general case with an arbitrary data set. Second, which singular question should the classification seek to answer by training said algorithm? Current research in this field is implicitly limited by forcing the researchers to select a question pertaining to their area of interest, select a set of features they predict may be informative, then select one or more of the multitude of machine learning algorithms to train and test their hypothesis. This method severely limits the potential impact the study will have due to the unpredictable nature of pairing algorithms with features to find the pairing that produces the best results. Impact is further limited by the need to clearly define a single question or aspect of the data to explore using machine learning, limiting the scope of the study.

At the abstract level, there exists a problem domain that can be described as the independent classification of multiple research questions within the same contextual background. This domain extends throughout multiple subject domains, allowing for potential multidisciplinary research and applications to expand multiple knowledge bases. This characterization of the problem domain as a classification problem naturally leads to a machine learning solution. However, a mathematically valid method of pre-determining which machine learning algorithm will produce the best results continues to elude researchers. Thus, research in this problem domain must contend with the need to test multiple algorithms and compare the results, selecting one and labeling it the "best" of the tested sample. An additional complication is the lack of a valid method to predict which set of features matched with that algorithm will lead to that best result. Therefore, testing must involve a variety of feature set and algorithm combinations if the goal is not only to find a solution that is "good enough" but also a solution that is empirically shown to be better than all other tested solutions. Testing methodology that allows for such extensive testing of combinations would also be inherently flexible in its applicability to a variety of subject domains. Superimposing another layer onto this methodology also allows multiple, independent classifications to be performed by the best-tested solutions.

METHODOLOGY: MACHINE LEARNING FRAMEWORK

To find an automated solution for localizing the seizure onset zone in epilepsy patients, previous studies have explored applying machine learning to neuro-imaging data with limited success. These include attempting to analyze MEG data with a support vector machine to identify high frequency oscillations thought to correspond to epilepsy activity (Guo et al, 2018). Others have used the support vector machine to analyze graph theoretical features extracted from fMRI data to lateralize and localize seizures to the temporal lobes (Wu et al, 2018. Others attempted to train an artificial neural network and support vector machine on frequency domain features extracted from MEG data to discriminate between healthy and controls and epilepsy and between frontal focal epilepsy and generalized epilepsy (Aoe et al, 2019; Soriano et al, 2017). These studies are representative of a large portion of articles I have read over the years because of their use of either the support vector machine or artificial neural network. While they often do not discuss how they arrived at this algorithm, we suspect they chose it because they saw it used with some success in previous literature. This represents an almost arbitrary decision in their methodology. Our proposed framework seeks to correct this aspect of the literature by providing a more comprehensive approach toward predictive model construction and selection. Early results from our framework include some models with accuracy consistent with these previous studies, but also some models with higher accuracy, sometimes using the same classifier but with different features.

To remove the limitations seen in previous studies when applying machine learning techniques, we first developed a machine learning framework intended to encompass the end-to-end developmental lifecycle of machine learning solutions. In doing so, we also realized that such a framework could also be utilized as the basis for an introductory course in machine learning. We provide that paper here for reference.

A Conceptual Framework for an Introductory Machine Learning Course

ANTHONY D. BOWMAN, LEON JOLOLIAN

Journal of Computing Sciences in Colleges

ABSTRACT

Historically, computer science curricula have largely focused on the development of algorithmic solutions. However, in recent years a new paradigm has emerged which focuses on machine learning as a data-driven approach to problem solving. To better equip students with the background needed for this emerging method of problem solving, the curricula would benefit from including a greater emphasis on the concepts frequently found in a data-driven workflow. Furthermore, continued advances in hardware should be considered to exploit parallelism in computations. Other concepts from machine learning include data cleaning, feature engineering, and concurrency in computation. In this paper, we propose a conceptual framework for an introductory course in machine learning with a data-driven workflow. With this framework, students will be exposed to the high-level design and concepts of data-driven development, gradually equipping them with the tools to fully grasp the machine learning paradigm. Further research is ongoing to construct a development environment that supports this framework for full adoption of the machine learning paradigm.

1 Introduction

Since their inception in the latter half of the 20th century, computer science curricula have faced the challenge of preparing students for the demands placed on them by their post-collegiate career paths. For the better part of that history, the predominant method of problem solving involved the development of algorithmic solutions, therefore curricula were adopted with courses teaching concepts and skills which supported such development including traditional and more recent models for software development ([1],

[10]). In recent years, new data-driven approaches like machine learning have emerged and continued to gain widespread prominence and complexity in both industry and academic research [5]. While there is significant overlap in the background knowledge required for both approaches, there are also some concepts in the data-driven workflow that bear greater emphasis because of the critical role they play in generating solutions. Thus, there is a need for curricula to adapt to the increasing demand for students with sound knowledge bases of both the algorithmic and data-driven approaches. To facilitate this, we propose a framework as the basis for an introductory course to the data-driven approach which will expose students to key concepts while providing a high-level design overview of the processing pipeline. Previous literature has seen proposals for degree programs in data science, including a range of courses to establish the knowledge base students will need to fully grasp this new paradigm [7]. A course built from our proposed framework can serve as the anchor course to such a program, providing students with a broad overview and context to tie in lessons learned in the rest of their coursework. Absent such a specialized degree program, this course would serve as the gateway course in a machine learning or data science concentration within a computer science or engineering degree program.

2 Machine Learning Development Framework

We characterize our framework by three layers of abstraction, shown below. Due to the more exploratory nature of the data-driven approach, it is natural to consider the developer as a researcher in this process. Doing so leads the student to shift their mindset more towards a research perspective where they must be asking and answering questions related to the data and the overall problem at multiple levels. This role would also lend itself to prepare students for an interdisciplinary team setting, increasingly common with the globalization of industrial markets [3]. While working in such a group environment, the student developer needs to be cognizant of the variety of engineering factors to be analyzed and communicate with other, potentially non-engineering team members [4]. Coursework and software to facilitate this team setting and interaction have been created with positive results in previous literature ([2], [8]). Effective communication in this setting is a critical skill that can be applied in the industrial setting while working with a client to explore and clarify their needs, which often must be translated from domain specific vernacular to the engineering domain. Thus, our framework for an introductory course in data-driven methodology includes ample opportunity for the students to interact with and gather feedback from others at each layer.

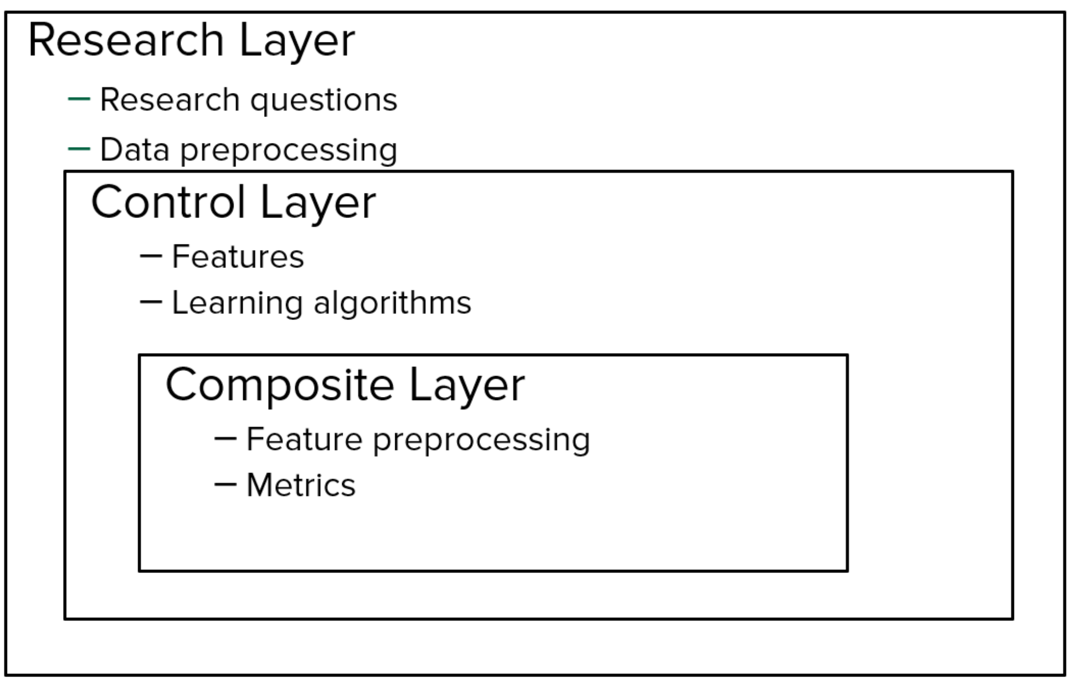


Figure 1: High Level Design

Beginning with a high-level view, our framework guides the researcher through multiple layers of abstraction encompassing the machine learning processing pipeline. The traditional mindset and pipeline as seen in previous literature commits the researcher to a particular solution, locking them into that specific implementation within the vast solution space, often without justification. This unnecessary restriction often leads to sub-optimal results due to current limitations in valid methods of predicting the optimal algorithm [12]. Our framework addresses this problem directly by forcing the researcher to consider alternatives at each layer, guiding them to more thoroughly explore that solution space. At the highest level of abstraction, the research layer forces the researcher to consider the possibility that multiple research questions could be tested using the same data set, perhaps exploring the same research topic but from different perspectives. Preprocessing of the data in different ways could also be considered in its potential to influence the results of the research. Descending into the control layer, the researcher would then be directed toward possible feature sets and subsets thereof before generating a set of possible machine learning algorithms for testing. Students would be instructed as to how this layer would also provide guidance in how to parallelize the process of building and testing the resulting models, leading into the composite layer. Here, the researcher would be faced with the variety of ways the features can be preprocessed before model construction. Different metrics in testing methodology should also be considered as related to the research questions decided upon at the top level.

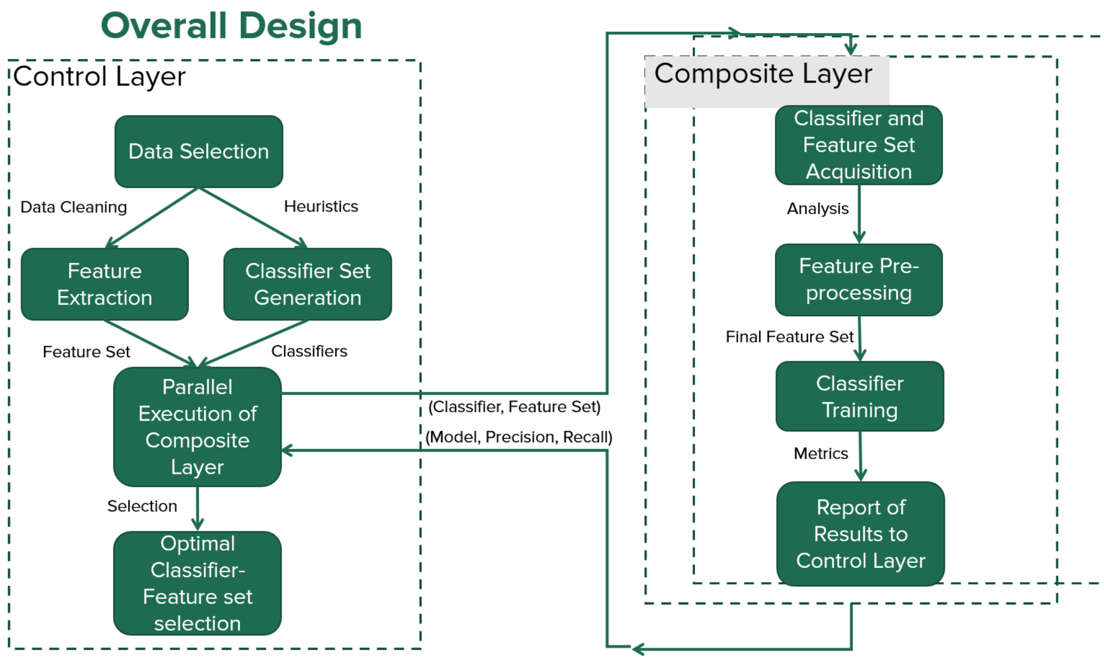


Figure 2: Control and Composite Layer Internal Structure

Establishing a three-layer architecture can naturally be thought of in a hierarchical way: The lower, control layer involves the aforementioned testing methodology, which is constructed as a set of classification problems. For every classification problem, the layer includes a user-defined feature set and set of algorithms. Every algorithm is then trained and tested with a variety of feature subsets. One pairing of an algorithm with feature subset is selected based on performance metric(s) and deemed the empirically best solution to the current classification problem. The control layer repeats this process for all classification problems until it arrives at a set of best solutions for those problems.

This solution set is contained within the control layer, which encapsulates them into a single package that is prepared to accept an unknown instance from the subject domain. The control layer extracts the appropriate features from this instance of interest and provides them to the package, distributing the features to each trained algorithm according to the algorithm-feature pairings. Each algorithm in this ensemble then produces a classification independent of each other with the complete set of classifications provided to the user afterward. This overall approach allows for a great degree of flexibility as each algorithm in the ensemble can adapt to the needs of each problem. The problems themselves can also be complementary in the subject domain or completely independent, allowing for further flexibility in adapting to the needs of the domain user.

3 Summary

With the increasing prevalence of data-driven problem solving, computer science curricula stand to benefit from including additional coursework specifically geared for that approach. In this paper, we proposed a conceptual framework to form the basis of a course for data-driven problem solving which would introduce them to critical concepts and provide a high-level design. Through this course, the curricula can show students how to adopt the machine learning paradigm and serve as students' jumping off point into the world of big data, machine learning, and the like.

4 Future Work

Continual development of this framework is ongoing to establish it as the conceptual design of a development environment built from the ground up with data-driven methodology in mind. In addition, we have tested this framework as a machine learning model applied to a case study involving the localization of the seizure onset zone within the brain of epilepsy patients. Early results show great promise when compared to the limited success seen by previous studies [6]. Preliminary results show higher accuracy and the ability of our framework to facilitate the application of data-driven research for parallel research projects. In the interim, our framework can serve as both a conceptual design for our proposed introductory course into machine learning and data-driven problem solving as well as a high-level model for students to follow in future projects to facilitate full adoption of the machine learning paradigm. This embracing of the machine learning paradigm would be further enhanced by using a development environment constructed by implementing our proposed framework. Thus, our framework can serve as the basis for the course to teach students, a model for students to follow, and the underlying architecture for the tool students can use for their machine learning needs.

References

[1] Beck, K., and Fowler, M. (2001). *Planning extreme programming* (Vol. 200). Reading: Addison-Wesley.

[2] Burnell, L. J., Priest, J. W., and Durrett, J. B. (2002). Teaching distributed multidisciplinary software development. *IEEE software*, *19*(5), 86-93.

[3] Duderstadt, J. J. (2010). Engineering for a changing world. In *Holistic engineering education* (pp. 17-35). Springer New York.

[4] Ertas, A., Maxwell, T., Rainey, V. P., and Tanik, M. M. (2003). Transformation of higher education: the transdisciplinary approach in engineering. *IEEE Transactions on Education*, *46*(2), 289-295.

[5] Eyigoz, E., Mathur, S., Santamaria, M., Cecchi, G., and Naylor, M. (2020). Linguistic markers predict onset of Alzheimer's disease. *EClinicalMedicine*, *28*, 100583.

[6] Guo, J., Yang, K., Liu, H., Yin, C., Xiang, J., Li, H., ... and Gao, Y. (2018). A stacked sparse autoencoder-based detector for automatic identification of neuromagnetic high frequency oscillations in epilepsy. *IEEE transactions on medical imaging*, *37*(11), 2474-2482.

[7] Bile Hassan, I., Ghanem, T., Jacobson, D., Jin, S., Johnson, K., Sulieman, D., and Wei, W. (2021, March). Data science curriculum design: A case study. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (pp. 529-534).

[8] Jaccheri, L., and Sindre, G. (2007, July). Software engineering students meet interdisciplinary project work and art. In *Information Visualization, 2007. IV'07. 11th International Conference* (pp. 925-934). IEEE.

[9] Jukic, S., Saracevic, M., Subasi, A., and Kevric, J. (2020). Comparison of ensemble machine learning methods for automated classification of focal and non-focal epileptic EEG signals. *Mathematics*, *8*(9), 1481.

[10] Randell, B. (1996). The 1968/69 NATO software engineering reports. *History of software engineering*, *37*.

[11] Royce, W. W. (1987, March). Managing the development of large software systems: concepts and techniques. In *Proceedings of the 9th international conference on Software Engineering* (pp. 328-338).

[12] Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, *1*(1), 67-82.

Framework Evaluation

To judge quantitatively and qualitatively judge the performance of this framework, we consider the standard statistical metrics of precision, recall, and f-measure for evaluating a trained machine learning model. Doing so allows us to compare the final output of each lower layer iteration to results from a study using the standard machine learning approach i.e. training and testing a single machine learning algorithm with a single feature set. In this comparison, the lower layer of the framework could be considered an expansion of the standard testing methodology by including the single algorithm within its list of tested algorithms and the single feature set would be included as a subset of the feature set provided to the algorithm set. If we take this situation to be representative of the average case, then a direct comparison to the output of the lower layer yields an expected range of possible measured differences.

In the worst case, the framework's composite layer precisely matches the performance metrics of the standard approach, thereby confirming that particular model as the best performing mode tested. This only occurs when the model generated by the standard approach happens to be the best combination of algorithm and feature set selected from the framework's lists. We believe the likelihood of this instance to be quite low due to the ever-expanding list of possible algorithms and features that may be tested, especially as the feature set size increases.

In the best case, the framework's lower, composite layer yields a model with higher precision, recall, and f-measure than the standard approach to an arbitrary degree. Because the performance of the standard approach cannot be precisely predicted, its range of each metric is [0.0, 1.0]. If we take the average case to be 0.5 for each metric, then the best case benefit from the lower layer is 0.5, achieving perfect accuracy in each metric. We believe this instance is also unlikely, so we consider instead the potential gain to be in the range [0.0, 0.5] where a benefit of 0.0 is the worst case discussed above. Thus, the proposed lower layer of the framework may yield results that, at best, are substantial improvements to the standard approach, and are, at worst, exactly equal and thereby confirm the standard approach to be empirically the best.

Evaluation of the control layer of the framework requires consideration of the potential benefits gained from modeling multiple problems in parallel. We believe attempting to assign quantitative measures to these benefits would result in arbitrary units of measurement; therefore, we adopt a qualitative approach. The potential value of the overall framework can be directly compared to performing multiple, independent research studies (the lower layer iterations) followed by another study combining the best results of those into a composite with an over-arching theme specific to the subject domain. In this way, the framework may dramatically reduce the time cost and improve the model's generality and scientific validity by ensuring all work is carried out by the same researchers following the same protocol with data gathered with the same equipment.

Evaluation of the performance gained by incorporating advancements in parallel processing into the control layer can be done by analysis of its computational efficiency. Analysis of the traditional workflow shows a quadratic efficiency where the time requirement to completion increases in relation to the number of classifiers and features, or $O(n^2)$ behavior. Parallelizing the training of predictive models in our framework reduces this time requirement down to $O(n)$ or linear complexity. More precisely, the parallelized framework scales according to O(C + F) where C and F are the size of the classifier and feature sets respectively. We include our published paper detailing this machine learning framework here.

A Framework for an Automated Development Environment to Support the Data-driven Machine Learning Paradigm

ANTHONY D. BOWMAN, SHYAM P. PRABHAKAR, LEON JOLOLIAN

SoutheastCon 2022, pp. 329-331

Format adapted for dissertation

Abstract—In recent years a machine learning paradigm has emerged, focusing on a data-driven approach as opposed to traditional development. Advances in machine learning techniques have allowed researchers to make substantial gains in tackling complex problems in diverse fields such as medical diagnosis through image analysis, object detection and tracking, and natural language processing. However, often researchers only employ one or two machine learning algorithms with a static feature set while only testing a single hypothesis. This self-imposed bottleneck often produces suboptimal results because it arises from using machine learning within the classical, algorithmic context using existing development tools. Therefore, there is a need to create new development tools which reflect this change to the machine learning paradigm. In this research, we propose a development environment that allows researchers to leverage those capabilities more fully by shifting not only the tool they use but also their mindset. Our proposed environment serves as an intermediate tool, guiding the researcher and making full adoption of the machine learning paradigm throughout the software development process easier. To accomplish this, our framework is defined by a three-layer structure designed for subject domain assessment, data manipulation and feature set exploration. Supported by parallelism, data cleaning and feature engineering, this research provides a conceptual basis for future creation of development environments for the machine learning paradigm. Future development of such a conceptual design would allow for additional intelligent tools to aid the user in designing solutions and support reusability at the design level.

Keywords—development environment, machine learning, software development

Introduction

Advances in machine learning techniques have allowed researchers to make substantial gains in tackling difficult problems in diverse fields such as medical diagnosis, image recognition, and natural language processing. However, often, machine learning has been used as a blunt instrument that is designed to ingest large amounts of raw data and produce usable predictive models that can help in decision making. Committing substantial resources to power this machine learning tool may still produce subpar results due to a dilemma inherent to machine learning: The combination of classification algorithm and feature set that will produce the best results cannot currently be predicted for any arbitrary data set or research question [1]. This problem then encourages the researcher to explore the solution space to compare models generated by different learning algorithms trained with a variety of features. However, research is often published which only focuses on a model generated by a single machine learning algorithm, trained by a static data set without justification as to why specifically that algorithm was selected. We believe this to be a consequence of the researchers' inexperience with machine learning as its prevalence increases, permeating fields beyond engineering such as medical research [2]. Furthermore, research papers employing machine learning methodology are often focused on answering a single research question. We believe this to be a natural consequence of the historical focus on research using the scientific method where only a single hypothesis should be tested at a time. While there is, of course, still cause to utilize such a focused approach, this limitation need not be applied when exploring using data-driven methodology.

A survey of the literature using machine learning shows several commonly used development environments such as MATLAB and Jupyter Notebook [3 - 8]. From this review, we generated a list of attributes a development environment would ideally incorporate into its design for data-driven research.

These include support for the researcher to pursue multiple research questions in parallel, data selection and cleaning, feature engineering with dimensionality reduction, selection of multiple machine learning algorithms if desired with parallelized algorithm training and the ability to define and view a selection of testing metrics. This environment should also support the reuse of features and data across multiple research projects when possible, improving the efficiency of research conducted within this environment. The ability to incorporate additional modules for domain specific applications and additional intelligent features to augment the above functions would further improve the environment's adaptability and ease of use. Comparing this list of requirements to those offered by commonly used environments, we see that while some features are supported, their support is often limited to various degrees, resulting in incomplete attribute sets with regards to the ideal environment for data-driven development. A partial view of this analysis can be seen in Table 1. We believe the limited implementation of these features is largely in part due to development environments such as MATLAB predating the advent of machine learning methodology, thus they were not designed with data-driven techniques in mind. These environments have been developed to be geared more toward classical approaches to software engineering and modern improvements with their focus on a singular outcome paralleling the focus on a single research question [9, 10]. With the requirements for a data-driven focus in mind, we will now introduce our proposed framework for such a development environment. We previously published this framework in the educational context as the basis for educating students about the machine learning processing pipeline, and now present the framework as the basis for a tool to aid in that education and future, widespread use [11]. We encourage the reader to review this previous work for additional, lower-level details of each layer within the framework.

Development environment framework

We define our environment with a modular, three-layer embedded structure to model and streamline the machine learning processing pipeline. Following this approach, the researcher is guided through a more comprehensive application of machine learning methodology to their subject domain, beginning with considering the possibility for multiple research questions in tandem. Defining feature and classifier sets followed by dimensionality reduction and parallel model construction provides a thorough search through the solution space with the parameters and results contained within an organized super-structure. This approach provides more optimal results and a clear avenue toward meta-analysis between the results of different research questions. The framework's object-oriented design also lends itself toward an easily upgradeable, modular implementation that allows integration of new, more sophisticated modules. In this way, the framework may easily evolve as our understanding of machine learning methodology advances.

TABLE I. COMMON DEVELOPMENT ENVIRONMENTS

	Matlab	Eclipse	Jupyter Notebook	Google Colab	VS Code	DataRobot
Data selection	L	L	L	L	L	L
Data preprocessing	H	H	H	H	H	H
ML Algorithms	H	H	H	H	H	H
Dimensionality reduction	L	L	H	H	H	H
Modules/Libraries	H	H	H	H	H	L
Specialization	L	H	L	L	L	L

To encourage adoption of the machine learning paradigm, we propose this novel framework for machine learning research that attempts to provide a systematic means of guiding and framing research progress. To accomplish this, we define a multi-layer approach: The middle Control layer attempts to address the unpredictability issue by testing multiple algorithms with a variety of feature sets, with the lower Composite layer generating performance statistics for each algorithm-feature set pairing. Performing this testing improves the ability of the study to select the pairing with the best results, at least of those tested. This layer is encapsulated within an upper Research layer, which can order the Control layer to repeat its process for multiple feature sets and instance labels. Doing so allows the Research layer to define multiple questions to be trained and tested by the lower layers, providing the potential to greatly expand the scope of the research to an arbitrary degree defined by the researchers as they define the research question(s) to be investigated.

The Control and Composite layers are contained within the Research layer which provides the higher-level context. During this layer, the researcher is guided toward considering alternate research questions that could be investigated simultaneously through parallel use of this environment. In this manner, the equivalent of multiple research projects can be undertaken while potentially reusing

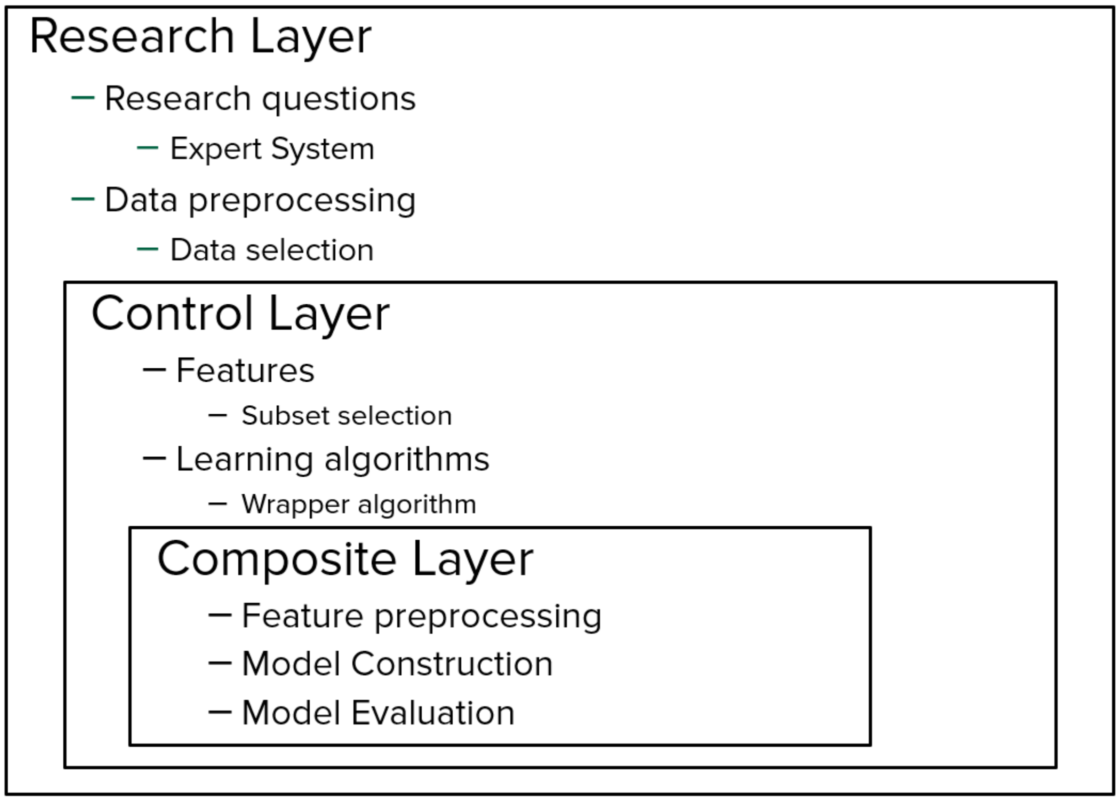


Fig 1. Conceptual Framework

some feature sets, reducing the time required.

Summary

While data-driven problem solving has become increasingly more common, machine learning methodology is often under-utilized through limited model exploration. In this paper we identified several attributes which a development should fully support to encourage greater adoption of the machine learning paradigm. We then presented a conceptual framework for such an environment characterized by three layers of abstraction. While we previously published this framework within the educational context for students' training in data-driven development, here we present the framework

as the basis for a development environment to guide developers and grow with them as our understanding of the machine learning paradigm evolves.

Future work

We continue to development this framework to incorporate more intelligent components within its design. These components will serve to further automate some functions while guiding the user toward full adoption of the machine learning paradigm at the conceptual level. In addition, we are looking to include features to improve and facilitate collaboration with versioning control functionality to maintain organization of projects at both the file and conceptual levels.

References

[1] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation,* vol. 1, no. 1, pp. 67-82, 1997.

[2] T. Wu, D. Chen, C. Qiqi, R. Zhang, W. Zhang, Y. Li, L. Zhang, H. Liu, S. Wan, T. Jiang and J. Zhang, " Automatic lateralization of temporal lobe epilepsy based on MEG network features using support vector machines," *Complexity,* no. Feb, 2018.

[3] MATLAB, version 9.4.0.813654 (R2018a), Natick, Massachusetts: The MathWorks Inc, 2018.

[4] J. Wiegand, "Eclipse: A platform for integrating development tools," *IBM Systems Journal,* vol. 43, no. 2, pp. 371-383, 2004.

[5] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay and P. Ivanov, "Jupyter Notebooks-a publishing format for reproducible computational workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, 2016, pp. 87-90.

[6] J. W. Johnson, "Benefits and Pitfalls of Jupyter Notebooks in the Classroom," in *Proceedings of the 21st Annual Conference on Information Technology Education*, 2020.

[7] J. F. Pimentel, L. Murta, V. Braganholo and J. Freire, "Understanding and improving the quality and reproducibility of Jupyter notebooks," *Empirical Software Engineering,* vol. 26, no. 4, pp. 1-55, 2021.

[8] E. Bisong, "Google Colaboratory," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, Berkeley, CA, Apress, 2019, pp. 59-64.

[9] K. Beck, M. Hendrickson and M. Fowler, Planning extreme programming, Addison-Wesley Professional, 2001.

[10] B. Randell, "The 1968/69 nato software engineering reports," *History of software engineering,* vol. 37, 1996.

[11] A. Bowman and L. Jololian, "A conceptual framework for an introductory machine learning course," in *Journal of Computing Sciences in Colleges*, Lacey, WA, 2021.

A Case Study in a Machine Learning Framework Applied to Epilepsy Localization

ANTHONY D. BOWMAN, WESLEY CONWELL, LEON JOLOLIAN

Submitted to IAJC 2022

Format adapted for dissertation

Abstract

The rise of machine learning methodologies in recent years has seen great success in a variety of applications. However, this new paradigm is often utilized in limited ways through arbitrary selection of machine learning algorithm(s) and static feature sets, particularly in the medical literature. We have previously published a framework that removes these artificial limiters while laying the groundwork for parallel research and development tracks. To showcase the potential power from this expanded use of the machine learning paradigm, we applied this framework to the complex medical problem of epileptic seizure localization. Resting state EEG/MEG data were simultaneously collected from 22 patients prior to epilepsy surgery and retroactively selected for analysis. Power spectral and coherence features were extracted from all sensor time series data. Sets and subsets of these features were used to train multiple machine learning algorithms for classifying epilepsy in different brain regions. Models generated by a variety of algorithms and trained by delta, theta, beta, and low gamma MEG and EEG features were able to achieve an f-measure > 0.95 when distinguishing between left frontal epilepsy and bilateral extra-frontal epilepsy patients. Results show the artificial neural network also achieved this f-measure, but only when trained on the subset of features including beta and low gamma EEG features. Models generated by training the same algorithms and feature sets only achieved maximally an f-measure of 0.818 when classifying right frontal epilepsy versus bilateral extra-frontal epilepsy. In this study, using parallel applications of the machine learning paradigm, we were able to both improve on results seen in previous studies in classifying epilepsy and showcase the

potential for meta-analysis across research tracks. This study provides additional insights into how research can be greatly expedited and expanded in scope through parallel exploration of topics which share overlapping feature or data sets.

Introduction

With the emergence of the machine learning paradigm in recent years, researchers in a wide variety of fields have sought out solutions generated by machine learning algorithms. While machine learning techniques continue to evolve, their adoption and usage is often carried out in an ad hoc manner, particularly by those whose expertise lies outside the field of computer science or related disciplines. One method to help bridge the gap between the non-expert and such an evolving methodology would be to introduce new development environments to aid in facilitating adoption of the machine learning paradigm. We have previously published a novel framework designed from the ground up with this specific purpose, first as a framework for a course in machine learning and later as a framework for a development environment (Bowman & Jololian, 2021; Bowman, Prabhakar & Jololian, 2022). We encourage the reader to review these papers for a detailed description of the framework. In this research, we applied this framework and methodology to a problem in the medical domain involving the localization of seizure onset zones in epilepsy patients.

To find an automated solution for localizing the seizure onset zone in patients with focal epilepsy, previous studies have explored applying machine learning to neuro-imaging data with limited success. These include attempting to analyze MEG data with a support vector machine to identify high frequency oscillations thought to correspond to epilepsy activity (Guo et al, 2018). Others have used the support vector machine to analyze graph theoretical features extracted from fMRI data to lateralize and localize seizures to the temporal lobes (Wu et al, 2018). Other studies attempted to train an artificial neural network and support vector machine on frequency domain features extracted from MEG data to discriminate between healthy and controls and epilepsy and between frontal focal epilepsy and generalized epilepsy (Aoe et al, 2019; Soriano et al, 2017). These studies are representative of a large portion of medical literature we have reviewed because of their use of either the support vector machine or artificial neural network. While they often do not discuss how they arrived at the algorithm used, we suspect they chose it because they saw it employed with some success in previous literature. This represents an almost arbitrary decision in their methodology. Our proposed framework seeks to correct this aspect of the literature by providing a more comprehensive approach toward predictive model construction and selection. Early results from our framework presented in this paper include some models with accuracy consistent with these previous studies, but also some models with higher accuracy, sometimes using the same classifier but with different features.

Applying our framework to the case study, we instantiated this three-layer architecture with each layer broadening the scope of our search through the solution space. Beginning from the research layer, we were able to generate multiple research questions that could be explored from the same data set. Data selection and preprocessing was then determined according to the needs of those research questions with one control layer generated to address each question. Within each control layer the feature and classifier sets were then defined, and feature extraction implemented. Of note is the ability for this architecture to allow for the researcher to explore the effect completely different features have in relation to each research question. Model construction and testing was then performed in every instantiation of the composite layer with a confusion matrix generated.

Methods

We retrospectively selected patients in our database with medically intractable epilepsy (n = 22). All patients had previously undergone surgical resection and were seizure free for at least six months, thus confirming their epileptic locus was within the resected region. As part of their pre-surgical evaluation, all patients had an MEG study performed using the system described below. The study was approved by the Institutional Review Board at the University of Alabama at Birmingham.

MEG Acquisition and Preprocessing

All MEG recordings were performed using a whole-head, 148-channel system housed within a magnetically shielded room (4D Neuroimaging, San Diego, CA). All patients were in a reclined position for the duration of the recordings. Multiple recordings were collected from each patient, each lasting 10 min and collected at a sampling rate of

508.63 Hz. Each data file was then preprocessed with in-house MATLAB scripts using Statistical Parametric Mapping software (SPM12b, http://www.fil.ion.ucl.ac.uk/spm/).

EEG Acquisition and Preprocessing

All EEG recordings were gathered using the International 10-20 system of electrode placement and gathered concurrently with each MEG recording. All EEG data was gathered at a sampling rate of 2 kHz and down-sampled to 600 Hz using a low-pass filter. Only data from 25 EEG leads common across all patients were included for further analysis and feature extraction.

MEG and EEG Spectral and Coherence Feature Extraction

Feature extraction from both MEG and EEG recordings was performed using a combination of Brainstorm functions and in-house MATLAB scripting (MATLAB, 2018; Tadel et al, 2011). Mean power spectrum density (Welch method) was computed using the Brainstorm function with the frequency bands slightly adjusted to the following: Delta band from 1 – 3 Hz, theta band from 3 – 8 Hz, alpha band from 8 – 12 Hz, beta band from 15 – 29 Hz, low gamma band from 30 – 59 Hz, high gamma band from 60 – 90 Hz. Spectral coherence features between sensors were computed from both MEG and EEG time series data using the mean square coherence function in MATLAB 2018a.

Machine Learning Training and Testing

All classifier training and testing was performed using the Waikato Environment for Knowledge Analysis (WEKA) open-source machine learning software (Witten et al, 1999). Default parameters were used for each classifier unless otherwise noted. Unless otherwise noted, all classifiers were trained and tested using 10-fold leave-one-out cross validation with WEKA reporting the confusion matrix, precision, recall, and weighted f-measure for each trained model. Classification was performed using different combinations of features provided to each classifier to find the combination of classifier and feature set that produced the highest f-measure.

Results

In Table 1, we present some results with seizure localization to the left frontal lobe. Using the weighted f-measure reported by WEKA as our metric for this matrix, we were able to see a wide range in results depending on algorithm and feature set. For this and future tables of results, we denote a calculated f-measure as "not a number" ("NaN") when the calculation involves division by zero. The maximum f-measure for each table is bolded. Maximum f-measure of 0.951 was achieved from models generated by logistic regression, stochastic gradient descent (SGD), simple logistic, support vector machine (SMV), and logistic model tree (LMT) algorithms trained by a feature set containing MEG power in the delta, theta, beta, and low gamma ranges. In the case of logistic regression and SVM, achieved maximum f-measure decreased when features extracted from EEG were included in addition to the aforementioned MEG features. The multilayer perception, otherwise known as the artificial neural network (ANN), was the only algorithm to match the maximum f-measure with fewer, albeit different features.

Table 1: Left frontal vs bilateral extra-frontal

Algorithm	Delta EEG	Theta EEG	Beta and low gamma EEG	Delta, theta, beta, low gamma EEG	Delta, theta, beta, low gamma MEG	EEG&MEG
BayesNet	NaN	0.777	NaN	0.777	0.872	0.872
NaiveBayes	0.653	0.753	0.777	0.777	0.291	0.347
NaiveBayesMultinomial	NaN	NaN	NaN	NaN	NaN	NaN
Logistic Regression	0.727	0.727	0.818	0.753	**0.951**	0.836
SGD	0.777	0.777	**0.909**	0.777	**0.951**	**0.951**
Multilayer Perceptron	0.777	0.777	**0.951**	0.777	**0.909**	**0.909**
SimpleLogistic	0.753	0.777	**0.889**	0.777	**0.951**	**0.951**
SMO (SVM)	NaN	0.777	NaN	NaN	**0.951**	**0.909**
DecisionStump	0.753	0.753	0.777	0.753	0.872	0.872
J48	NaN	0.753	**0.889**	0.786	NaN	0.786
LMT (log tree)	0.753	0.777	0.889	0.777	**0.951**	**0.951**
Random Forest	NaN	0.777	NaN	NaN	0.777	0.852
Random Tree	0.700	0.727	**0.909**	0.727	0.852	0.818

Table 2 shows the results from the same algorithms trained with the same feature sets, relabeled to data from patients with right frontal epilepsy. In this classification task, the highest weighted f-measure of 0.818 was only achieved by the logistic regression algorithm trained using the combined MEG feature set with power in all stated frequency bands.

Table 2: Right frontal vs bilateral extra-frontal

Algorithm	Delta EEG	Theta EEG	Beta and low gamma EEG	Delta, theta, beta, low gamma EEG	Delta, theta, beta, low gamma MEG	EEG&MEG
BayesNet	NaN	NaN	NaN	NaN	**0.805**	0.805
NaiveBayes	0.084	0.364	0.611	0.261	0.287	0.287
NaiveBayesMultinomial	NaN	NaN	NaN	NaN	NaN	NaN
Logistic Regression	0.570	0.613	0.398	0.600	**0.818**	0.636
SGD	0.590	0.590	0.745	0.636	0.778	0.600
Multilayer Perceptron	0.513	0.484	0.422	0.600	0.727	0.579
SimpleLogistic	NaN	0.566	NaN	NaN	0.566	NaN
SMO (SVM)	NaN	0.590	NaN	0.642	0.611	0.566
DecisionStump	0.513	0.455	0.590	0.590	**0.805**	0.805
J48	0.540	0.540	0.513	0.485	NaN	0.485
LMT (log tree)	NaN	0.513	NaN	NaN	0.566	NaN
Random Forest	0.540	0.579	0.566	0.540	0.642	0.745
Random Tree	0.438	0.600	0.485	0.441	0.611	0.745

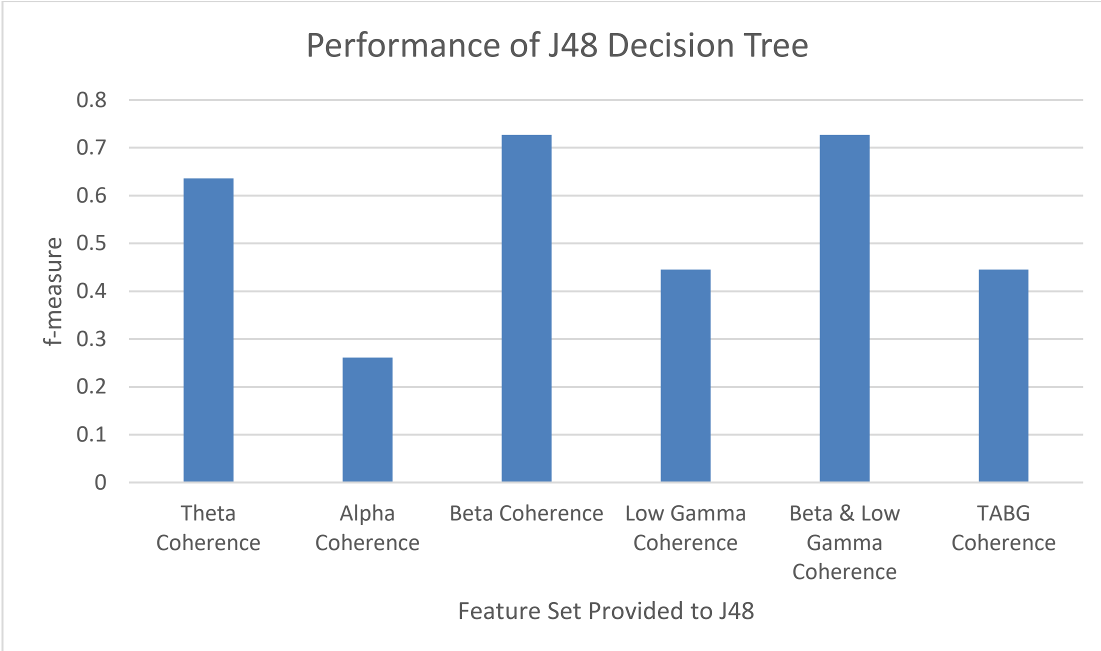
Table 3 shows the results training the same set of algorithms with coherence features extracted from MEG in the theta, alpha, beta, and low gamma frequency ranges (the far-right column being the combined feature set with all frequency bands included; "OOM" denotes an out-of-memory error when training the model). Here, the models were trained to discriminate between right and left temporal lobe epilepsy, also known as lateralization. In this task, the maximum weighted f-measure of 0.818 was achieved from the SimpleLogistic and LMT algorithms trained using only coherence features in the beta frequency band as well as the random tree algorithm using coherence features in the beta and low gamma bands.

Table 3: Left vs right temporal lobe epilepsy, MEG feature sets

Algorithm	Theta Coherence	Alpha Coherence	Beta Coherence	Low Gamma Coherence	Beta & Low Gamma Coherence	TABG Coherence
BayesNet	0.364	0.364	0.723	0.723	0.696	0.617
NaiveBayes	0.696	0.617	0.538	0.545	0.364	0.538
NaiveBayesMultinomial	0.091	0.636	0.364	0.455	0.538	0.445
Logistic Regression	0.445	0.331	0.723	0.445	0.636	0.455
SGD	0.331	0.261	0.538	0.636	0.723	0.723
Multilayer Perceptron	0.331	0.261	0.617	0.617	0.636	OOM
SimpleLogistic	0.140	0.195	**0.818**	0.445	0.727	0.331
SMO (SVM)	0.261	0.261	0.617	0.617	0.636	0.455
DecisionStump	0.331	0.140	0.727	0.445	0.727	0.331
J48	0.636	0.261	0.727	0.445	0.727	0.445
LMT (log tree)	0.140	0.195	**0.818**	0.445	0.727	0.331
Random Forest	0.331	0.364	0.808	0.455	0.696	0.636
Random Tree	0.445	0.455	0.723	0.455	**0.818**	0.538

Representing a single row in Table 3 in a different manner, Figure 1 below graphically shows how the performance of the J48 decision tree algorithm varies depending on which feature set was used. The performance of this algorithm does not improve with beta and low gamma coherence features combined instead of the subset including only coherence in the beta frequency range. Also noteworthy is the lower performance when trained on the full feature set including coherence features from all frequency bands.

Figure 1: Performance of J48 Decision Tree with various feature sets



Discussion of Technical Results

From an engineering standpoint, this methodology presents a more thorough exploration through both the solution space associated with each research question as well as the capability to efficiently investigate multiple research questions. Whereas the traditional methodology would have arrived at one or two of these models, the framework employed here allows for a much broader view of the landscape from which researchers can select the maximum weighted f-measure. Through this broader view, the authors see some models with tested f-measure approximately equal to those in previous studies, potentially confirming their results. Results also show some models achieve higher f-measure, sometimes using the same algorithm as previous studies but trained with a different feature set. In this case, the maximum was observed by multiple models which presents the researcher with the opportunity to consider other metrics as well as generate additional research questions that may explore why different feature sets resulted in high accuracy. Figure 1 visually shows the variability in performance of a single classifier, reinforcing the need for researchers to explore the solution space through multiple machine learning algorithms and feature sets. Further experimentation may explore the effect changing various hyper-parameters has on improving the performance of some algorithms, such as altering the number of hidden layers of the artificial neural network. Stepping back to take in a more abstract view of these results leads to additional questions relating to model performance relative to feature subset, in some cases decreasing substantially. This comparative meta-analysis is further expanded on by the opportunity to efficiently investigate parallel research questions (right and left frontal lobe epilepsy) and comparing the results from the same algorithms and feature sets. The authors were also afforded the opportunity to efficiently explore another research track in tandem by reusing the same data set, resulting in the investigation into the possible relationship between coherence features and focal epilepsy.

Discussion of Domain Specific Medical Results

From a medical standpoint, our results show an interesting disparity between the metrics achieved to discriminate between focal epilepsy in different brain regions. This is most easily seen when comparing the results in Tables 1 and 2: Investigating mirrored hypotheses, left frontal vs right frontal epilepsy, reveals classification of left frontal epilepsy using the defined power spectral features to be the “easier” of the two tasks. This may suggest the underlying neurophysiological characteristics associated with right frontal lobe epilepsy consist of a more complex pattern than that which can be identified by these algorithms and feature sets for left frontal lobe epilepsy. This may imply that focal epilepsy originating from the right frontal lobe may be better characterized by changes in features beyond power spectra. The nature of these features may provide further insight into the processing and functional structure of the right frontal lobe and how electrical pathologies such as epilepsy disrupt normal function.

Our results also show greater opportunity for success in such classification tasks with power spectral features over coherence features, although this observation may change with focal epilepsy in other cortical regions. Further research is needed with a larger data set to confirm this trend. Spectral coherence was chosen as a feature because of the connectivity exhibited between the temporal lobes and other regions of the brain (Haneef et al, 2014; Spencer, 2002). Previous studies have also explored other methods of quantifying neural or cortical connectivity such as transfer entropy, directed transfer functions, and graph theoretic metrics (Basu et al, 2015; Dai et al, 2012; Ursino et al, 2020; Wu et al, 2018). Mirroring the limited success seen in those studies, our results suggest further research is needed to explore these different metrics for connectivity in combination or refined, perhaps with more narrowly defined coherence metrics than used here. Our results clearly show the need to explore feature subsets to train multiple machine learning models for testing as a subset may lead to higher performance than the complete feature set.

Conclusions

In this study, we employed a previously published framework for the development of machine learning solutions conducted in the context of a medical case study. Using this methodology to embrace the machine learning paradigm more fully, we were able to efficiently explore both the problem and solution spaces within the case study's domain. Implementation of the composite layer allowed for empirical identification of algorithm-feature set pairings with higher performance metrics than seen in previous studies. Implementation of the control layer in combination of the composite layer greatly improved the scope of our work, expanding the potential of the project to explore multiple, related domain-specific questions in rapid succession through the re-use of data and feature sets.

Acknowledgements

This work was supported by an institutional grant from the University of Alabama at Birmingham. We would also like to acknowledge the contributions of Dr. Ismail Mohamed and Jeff Killen in data collection, management, and selection.

References

Aoe, J., Fukuma, R., Yanagisawa, T., Harada, T., Tanaka, M., Kobayashi, M., ... & Kishima, H. (2019). Automatic diagnosis of neurological diseases using MEG signals with a deep neural network. *Scientific reports*, *9*(1), 1-9.

Basu, I., Kudela, P., Korzeniewska, A., Franaszczuk, P. J., & Anderson, W. S. (2015). A study of the dynamics of seizure propagation across micro domains in the vicinity of the seizure onset zone. *Journal of neural engineering*, *12*(4), 046016.

Bowman, A. D., & Jololian, L. (2021). A conceptual framework for an introductory machine learning course. *Journal of Computing Sciences in Colleges*, *37*(1), 78-83.

Bowman, A. D., Prabhakar, S. P., & Jololian, L. (2022, March). A Framework for an Automated Development Environment to Support the Data-driven Machine Learning Paradigm. In *SoutheastCon 2022* (pp. 329-331). IEEE.

Cuffin, B. N., & Cohen, D. (1979). Comparison of the magnetoencephalogram and electroencephalogram. *Electroencephalography and clinical neurophysiology*, *47*(2), 132-146.

Dai, Y., Zhang, W., Dickens, D. L., & He, B. (2012). Source connectivity analysis from MEG and its application to epilepsy source localization. *Brain topography*, *25*(2), 157-166.

England, M. J., Liverman, C. T., Schultz, A. M., & Strawbridge, L. M. (2012). Epilepsy across the spectrum: Promoting health and understanding: A summary of the Institute of Medicine report. *Epilepsy & Behavior*, *25*(2), 266-276.

Guo, J., Yang, K., Liu, H., Yin, C., Xiang, J., Li, H., ... & Gao, Y. (2018). A stacked sparse autoencoder-based detector for automatic identification of neuromagnetic high frequency oscillations in epilepsy. *IEEE transactions on medical imaging*, *37*(11), 2474-2482.

Haneef, Z., Lenartowicz, A., Yeh, H. J., Levin, H. S., Engel Jr, J., & Stern, J. M. (2014). Functional connectivity of hippocampal networks in temporal lobe epilepsy. *Epilepsia*, *55*(1), 137-145.

MATLAB. (2018). *9.4.0.813654 (R2018a)*. Natick, Massachusetts: The MathWorks Inc.

Soriano, M. C., Niso, G., Clements, J., Ortín, S., Carrasco, S., Gudín, M., & Pereda, E. (2017). Automated detection of epileptic biomarkers in resting-state interictal MEG data. *Frontiers in neuroinformatics*, *11*, 43.

Spencer, S. S. (2002). Neural networks in human epilepsy: evidence of and implications for treatment. *Epilepsia*, *43*(3), 219-227.

Tadel, F., Baillet, S., Mosher, J. C., Pantazis, D., & Leahy, R. M. (2011). Brainstorm: a user-friendly application for MEG/EEG analysis. *Computational intelligence and neuroscience*, *2011*.

Tovar-Spinoza, Z. S., Ochi, A., Rutka, J. T., Go, C., & Otsubo, H. (2008). The role of magnetoencephalography in epilepsy surgery. *Neurosurgical focus*, *25*(3), E16.

Witten, I. H., Frank, E., Trigg, L. E., Hall, M. A., Holmes, G., & Cunningham, S. J. (1999). Weka: Practical machine learning tools and techniques with Java implementations.

Ursino, M., Ricci, G., & Magosso, E. (2020). Transfer entropy as a measure of brain connectivity: A critical analysis with the help of neural mass models. *Frontiers in computational neuroscience*, *14*, 45.

Wu, T., Chen, D., Chen, Q., Zhang, R., Zhang, W., Li, Y., ... & Zhang, J. (2018). Automatic lateralization of temporal lobe epilepsy based on MEG network features using support vector machines. *Complexity*, *2018*.

Biographies

Anthony D. Bowman is a doctoral candidate seeking a PhD in Computer Engineering at the University of Alabama at Birmingham. He earned both his MS in biomedical engineering in 2017 and BS in computer and information sciences in 2012 from the University of Alabama at Birmingham. Anthony D. Bowman's research interests include machine learning, data science, and bioinformatics. Anthony D. Bowman may be reached at anbowman@uab.edu.

Wesley Conwell is a PhD candidate of interdisciplinary engineering at the University of Alabama at Birmingham. He earned his BS in electrical engineering from the University of Alabama, MBA from the University of Alabama at Birmingham, and MS in electrical engineering from the University of Alabama at Birmingham. Mr. Conwell's research interests include smart cities, data analytics, big data, Internet of Things, and machine learning. Mr. Conwell may be reached at wlconwel@uab.edu.

Leon Jololian is a professor of electrical and computer engineering at the University of Alabama at Birmingham. He earned his BS in electrical engineering from Manhattan College, his MS in electrical engineering from Georgia Institute of Technology, his MS in computer science from Polytechnic University, and his PhD in computer science from the New Jersey Institute of Technology. Dr. Jololian's research interests include machine learning, mobile computing, and the internet of things. Dr. Jololian may be reached at leon@uab.edu.

METHODOLOGY: ADAPTED SPIRAL MODEL

We include here a copy of our manuscript submitted for publication in the ISEC 2022 special issue of *Journal of Systems and Software.* This manuscript includes a detailed description of our adapted spiral model as well as a walkthrough applying our model to the medical case study described above.

An Adaptation of the Spiral Model for the Integration of Hypothesis- and Data-driven Workflows

ANTHONY D. BOWMAN, LEON JOLOLIAN

Submitted to *Journal of Systems and Software*

Format adapted for dissertation

Abstract

Over the past fifty years a variety of methodologies have been created to aid developers in dealing with the complexities of software development. While methodologies such as agile development and the waterfall model have allowed for the tackling of increasingly demanding software solutions, they were not designed to efficiently handle data-driven development such as the novel paradigm of machine learning that has emerged in the last decade or so. With its rapid increase in prominence and different programmatic requirements than traditional software, the creation of dedicated development strategies is warranted to address this growing need. In this research we present an adaptation of the spiral model with the flexibility to accommodate both traditionally hypothesis-driven and emerging data-driven efforts such as machine learning. We then present results from a medical case study where we applied this updated lifecycle model, including a walkthrough through development to elucidate its advantages. We follow up with a discussion on the benefits our model provides for both software engineering and the specific medical problem domain. These include the potential rapid expansion of research in scope while enjoying accelerated results, particularly in parallel research tracks which share data and/or feature sets.

Keywords: Machine learning, data-driven research, software development

Introduction

With the recent advent of the machine learning paradigm, software engineers are faced with the challenge of developing increasingly complex solutions. These solutions can vary greatly in their requirements, particularly in their innate characteristics stemming from the problem domain. If the problem and solution are both well understood, then the traditional approaches to software engineering are well suited to constructing those solutions. This is largely due to those methodologies having a largely algorithmic nature (Randell, 1996; Royce, 1987). Even more modern approaches such as agile methodology, while providing greater flexibility in the developmental workflow, are still designed for classical problems which call for a single, target software product (Beck et al, 2011; Beck & Fowler, 2001). The rise of data-driven methodologies such as machine learning represents a paradigm shift in the creation of the solution i.e. a much more exploratory process versus the direct, algorithmic drive toward a clear target. This new paradigm also places a greater emphasis on additional tasks in data collection and preprocessing necessary for this methodology to produce valid solutions. This shift in programmatic requirements for data-driven methodologies necessitates a corresponding shift in the approach to developing those solutions and in the tools available to support such development. However, the increased prominence of data-driven solutions such as machine learning does not preclude the existence of problems which the traditional approaches of software development remain well suited to solving. Therefore, a new lifecycle model is needed which incorporates both data- and hypothesis-driven development of solutions as available options to the developer. Furthermore, a development environment designed to facilitate the workflow of such a lifecycle model would provide additional benefits to the developer through its adaptability and supporting features.

Using these criteria in our software engineering lab in the Department of Electrical and Computer Engineering at the University of Alabama at Birmingham, we have developed ideas to deal with this problem, resulting in the evolution of an existing software development model. By modifying the prominent spiral model previously published by Boehm in 1988, we allow for the model to adapt to the needs of the developer according to the demands of the project. If the project presents a problem which traditional development approaches are well suited for, the model guides the developer down that path. Likewise, if data-driven development is called for, there is an alternate path through the model which facilitates such development. However, should the developer ever deem a change is called for because their needs have shifted from one approach to another, then such a shift is also supported within this new model. Thus, the model allows for the development process to adapt to a change in the requirements of the developer. Predominant development environments available today are largely designed with either the traditional or novel data-driven development in mind. Common environments such as MATLAB which were constructed for the traditional approach are not as efficient when using their machine learning features which were added on later in their lifetime (MATLAB, 2018). On the other end of the spectrum, specialized environments such as AutoAI provide developers with the ease of accessibility to machine learning techniques (Wang et al, 2020). However, these environments offer poor support for the traditional approach and are further limited in their adoption of the machine learning paradigm by their often single-threaded mindset, unnecessarily constricting data-driven research through a hypothesis-driven bottleneck. To remove this bottleneck, we have previously published a framework for a development environment that offers greater support of the machine learning paradigm, which is included within our adaptive lifecycle model (Bowman, Prabhakar & Jololian, 2022). A development environment which thereby implements this adaptive spiral model would then provide the developer with not only a tool to support the more efficient creation of a greater variety of solutions but would also facilitate the surge of machine learning by offloading the expertise and programming proficiency needed into the tool. This would enable the developer to be the domain expert, directly developing solutions instead of relying on interfacing with an engineer trained in machine learning techniques.

To test this modified spiral model, we collaborated with a domain expert to investigate the complex problem of epileptic seizure localization. The domain expert, a neurologist, was interested in the potential for machine learning to create an automated solution of localizing the seizure onset zone in patients with medically intractable focal epilepsy. As anti-epileptic medications are only effective in approximately 75% of patients, surgical resection is considered to remove the epileptic focus (Coolen et al, 2018; England et al, 2012). While previous literature has shown the efficacy of MEG in pre-surgical guidance, the common method of source localization using dipole modeling is resource intensive in terms of both computational cost and personnel hours (Englot et al, 2015; Pataraia et al, 2004). Previous studies have attempted to employ machine learning techniques to develop an automated method of localizing the epileptic focus to a specific lobe with limited success (Aoe et al, 2019; Soriano et al, 2017). A common thread among these and other studies is their limited application of machine learning methodology, unnecessarily restricting themselves to one or two classifiers, often the support vector machine or artificial neural network. Previous studies exclusively employing either the traditional hypothesis-driven or data-driven methods have also achieved limited success (Aoe et al, 2019; Basu et al, 2015; Dai & He, 2011; Elisevich et al, 2011). This represents a more abstract view of the literature, revealing a parallel thread: The increased prevalence of complex problems which would greatly benefit from both hypothesis- and data-driven efforts working in tandem, instead of researchers choosing to tackle the problem using only one approach. The seemingly arbitrary choice in classifier demonstrates the frequent use of data-driven methodologies like machine learning as tools employed in a hypothesis-driven manner by domain experts inexperienced with such methodologies. This also mirrors the hypothesis-driven bottleneck seen in adapting environments designed with the traditional development approaches in mind to data-driven development. Through this case study we intend to show how our adaptive spiral model can be deployed to address these bottlenecks by shifting the mindset of both the software engineer and the domain expert while formalizing the integration of hypothesis- and data-driven development.

Adaptive Spiral Model Description

The original spiral model defines the gradual development of software through an iterative process, visualized on a 2D Cartesian plane as a path that spirals outward from the origin (Boehm, 1988). As this path travels through each of the four quadrants, development progresses through certain activities with each progressive spiral outward representing another iteration through the development lifecycle. The original model broadly categorizes the activities within each quadrant: “Determine objectives” (Quadrant II), “Identify and resolve risks” (Quadrant I), “Development and Test” (Quadrant IV), and “Plan the next iteration” (Quadrant III). Within each quadrant, the model specifies various tasks performed as the path continues through each iteration.

These include requirements planning in Quadrant I, risk analysis and assessment in Quadrant II, the design, production, and testing of a prototype within Quadrant III and either the cycling back for another iteration or final release in Quadrant IV. In the latter case, the outward spiral path terminates with the release of the software product.

We begin our description of an adaptive spiral model by first migrating some activities from Quadrant IV to Quadrant I, namely the activities associated with active generation of code to produce prototypes. We also label the origin as Phase 0, involving the creation of the project and initial specifications describing the topic and objectives.

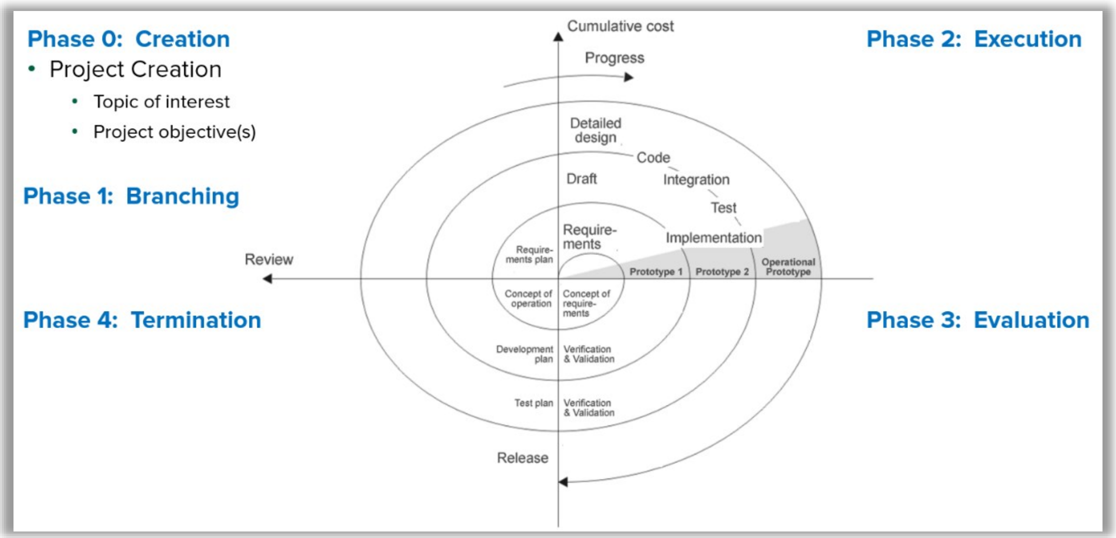


Figure 1: Adaptive Spiral Model Phase 0

The resulting adjustments are shown in Figure 1 with quadrants associated with a distinct phase until development leaves the spiral with Phase 4. Note that Phase 0 is not included after Figure 1 as it is never revisited throughout the subsequent iterations of the lifecycle.

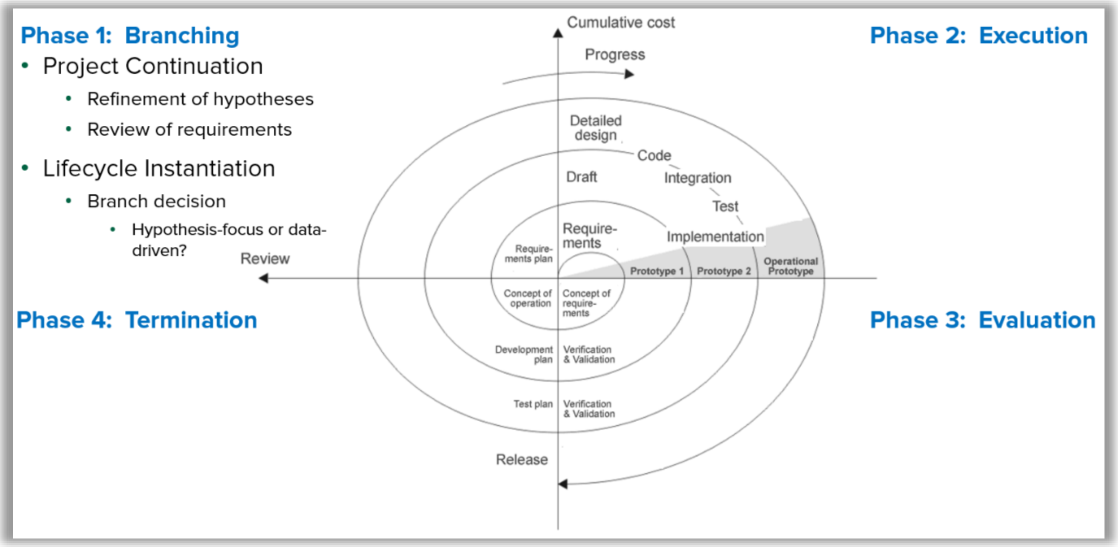


Figure 2: Adaptive Spiral Model Phase 1

Advancing through the model to Phase 1 takes the developer to the major branch decision in the model. After a review of the requirements for the project and any changes to the hypotheses, the developers are faced with a fork in the road: Would a traditionally algorithmic, hypothesis-driven approach or data-driven focus be better suited for the project to progress toward its goal(s)? Additional factors that may influence this decision include the availability of data suitable for a data-driven approach as well as a review of relevant literature showing how one approach was effective.

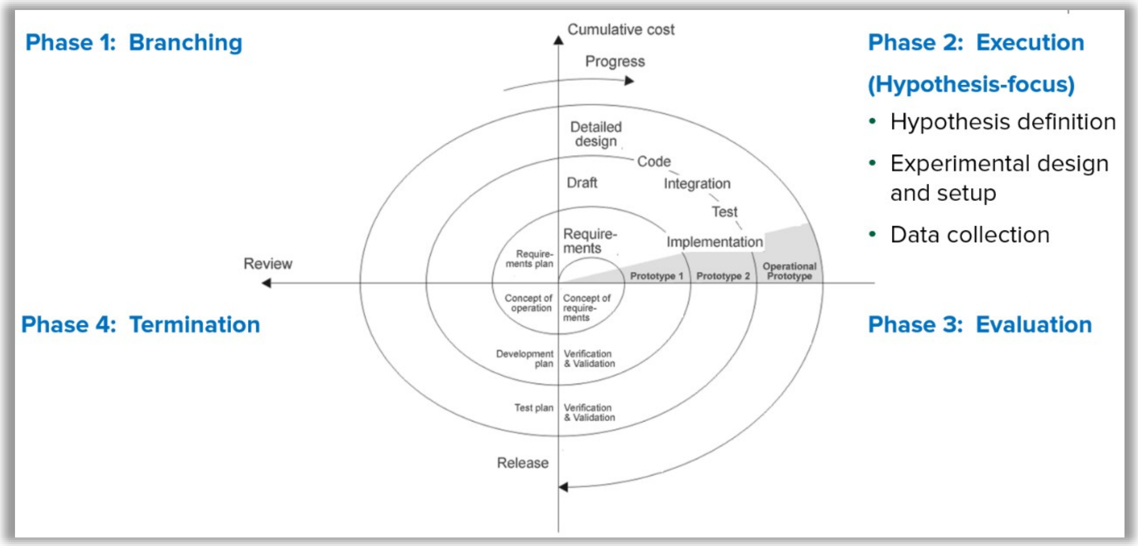


Figure 3: Adaptive Spiral Model Phase 2 with Hypothesis focus

If the developer chooses to focus on hypothesis-driven efforts, then the Execution phase involves the associated activities. A specific hypothesis is formulated and development progresses with that typically algorithmic focus in mind. Data is gathered and artifacts generated in a manner chosen by the developers that may include such approaches as extreme programming.

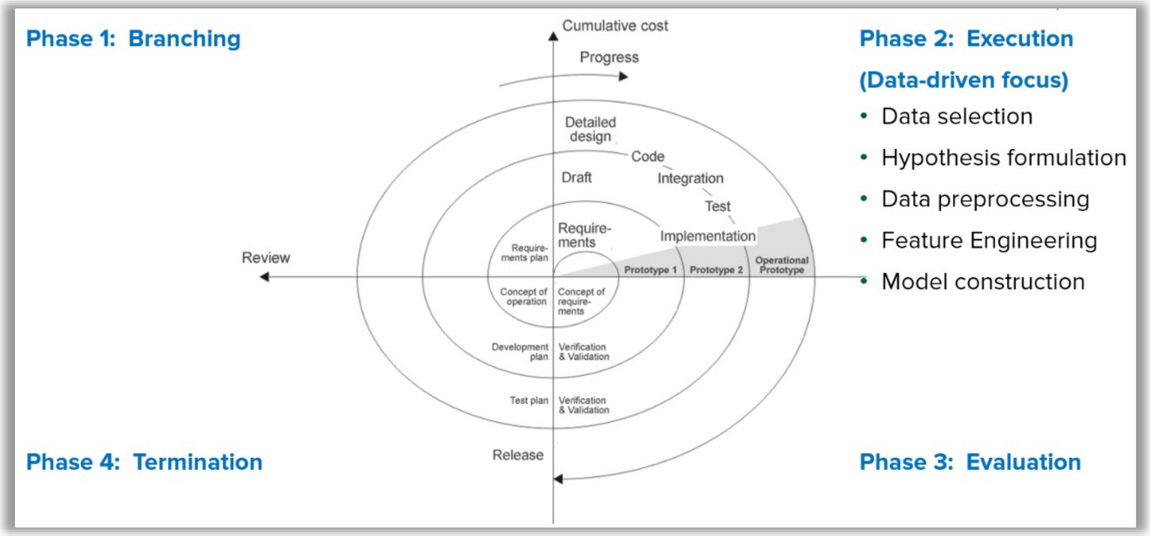


Figure 4: Adaptive Spiral Model Phase 2 with Data-driven focus

If the developer chooses to focus on data-driven efforts, then the Execution phase instead involves a different set of activities as shown in Figure 4. Development shifts to a mindset that is much more focused on data handling with a more lose hypothesis, allowing for later data analysis to guide future revisions to the hypothesis. More attention is given to how the data is selected, the quality of that data, and the techniques chosen for this iteration of the development cycle. For a more detailed description of a framework to facilitate greater adoption of the machine learning paradigm during this Phase, we encourage the reader to seek out our previously published research (Bowman & Jololian, 2021; Bowman, Prabhakar, & Jololian, 2022).

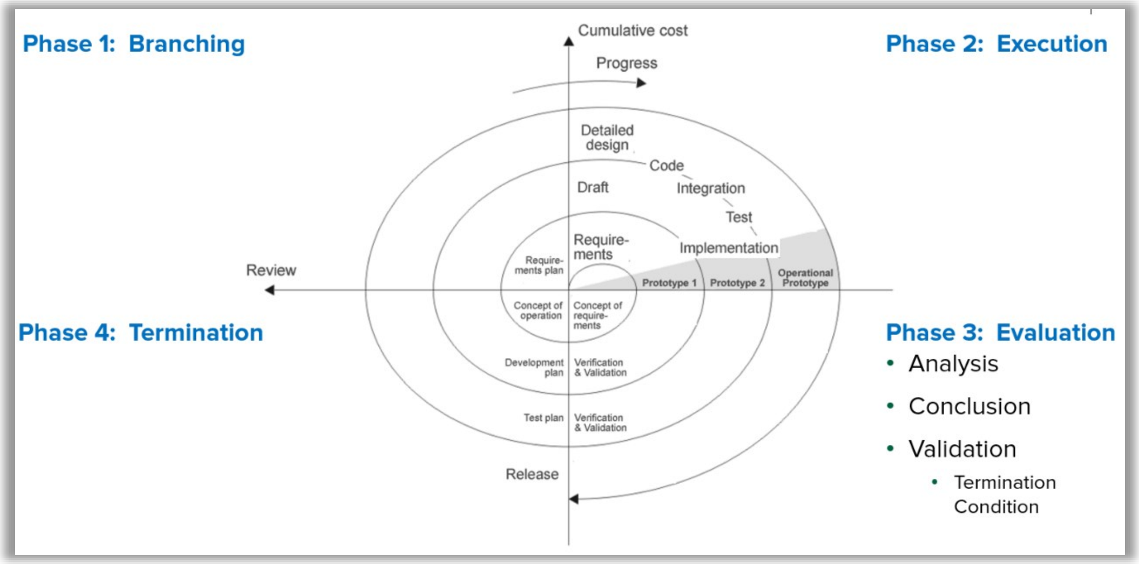


Figure 5: Adaptive Spiral Model Phase 3

Once the developer completes the execution phase, they progress into Phase 3 for evaluation. At this point, efforts are focused on analyzing the results of efforts from Phase 2. Results are summarized and conclusions drawn with the termination condition(s) for the overall project checked. If those conditions are not met, then the knowledge gained from this iteration of the lifecycle passes on to the next iteration with all artifacts.

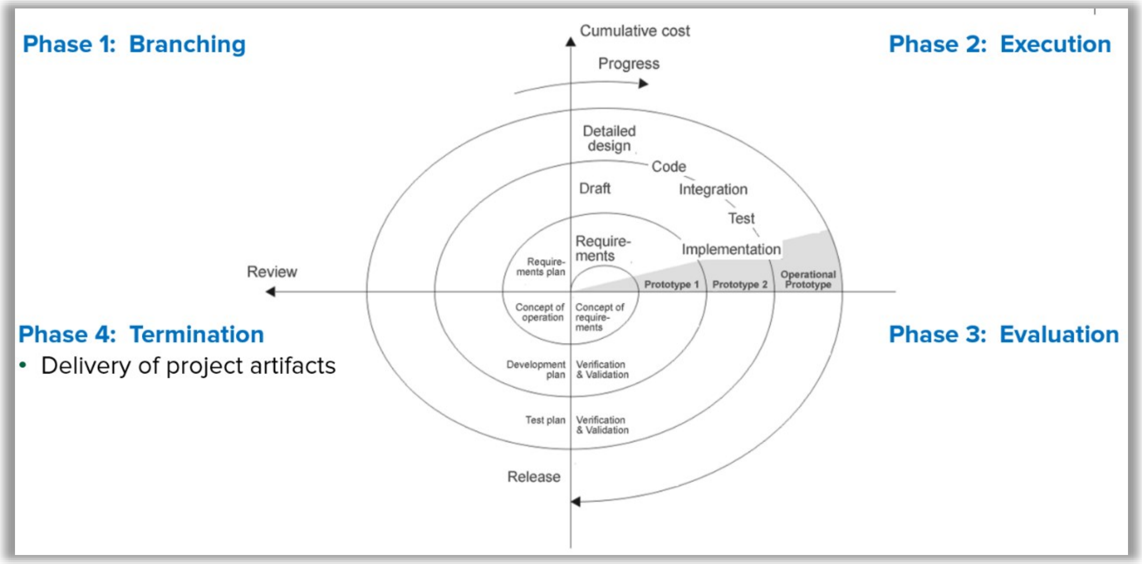


Figure 6: Adaptive Spiral Model Phase 4

Once it is determined that the termination condition has been met, development terminates in Phase 4. All project artifacts are delivered and/or handled according to prevailing best practices or agreed upon terms.

Case Study Methodology

We followed our proposed model while collaborating with the domain expert to research the question and posit ideas to investigate as possible solutions. For a detailed description of the methodology concerning patient data selection, preprocessing, and feature extraction, we encourage the reader to refer to our previous research (Bowman & Jololian, 2022). Patient data was retrospectively selected by the domain expert based upon the absence of seizures for at least 6 months after epilepsy surgery, thereby confirming the epileptic focus was in the brain region that was resected. All EEG and MEG recordings lasted 10 min and were collected at sampling rates of 2 kHz and 508.63 Hz respectively. The MEG data was recorded using a whole-head, 148-channel system housed within a magnetically shielded room (4D Neuroimaging, San Diego, CA). The EEG data gathered concurrently with each MEG recording was gathered using the International 10-20 system of electrode placement and subsequently down-sampled to 600 Hz using a low-pass filter. Data preprocessing and feature extraction for both EEG and MEG data were performed using the Brainstorm MATLAB toolbox and custom in-house MATLAB scripts (MATLAB, 2018; Tadel et al, 2011).

Not included in our previous paper is the exploration of time-invariant directed transfer functions (DTFs) to characterize behavior between sensors we explored later on and presented in this research (Dai & He, 2011). For that segment of this project, raw data was imported through the Brainstorm, formatted by a custom in-house MATLAB script and imported into the eConnectome toolbox for DTF calculations (He et al, 2011; Tadel et al, 2011). A multivariate autoregressive (MVAR) of fourth order was used, processing each 10 min run of MEG data in full within the 3 – 50 Hz frequency band. The surrogate method of statistical testing was employed with 50 permutations and a p value of 0.05 (Kamiński et al, 2001; Kugiumtzis, D., 2001; Theiler et al, 1991). All classifier training and testing presented in this research was also performed using the Waikato Environment for Knowledge Analysis (WEKA) open-source machine learning software (Witten et al, 1999). The primary metric reported in the tables below are the weighted f-measure as calculated by WEKA using default parameters for all classifiers.

Case Study Lifecycle Walkthrough

For this case study, project development was initiated with the goal of developing an automated solution for localizing the seizure onset zone in patients with medically intractable epilepsy. Thus, Phase 0 of our lifecycle model was completed with a clear topic of interest and objective in mind. Progressing into Phase 1, our broad hypothesis was narrowed to the investigation of lateralization of temporal lobe epilepsy using machine learning techniques to fulfill the stated objective, given their success seen in other problem domains and limited success seen in previous literature on this topic. Temporal lobe epilepsy was chosen because of the immediate availability of patient data as temporal lobe epilepsy is the most common form of focal epilepsy. Based on the choice of machine learning, the branch decision was clearly to proceed into the data-driven branch of Phase 2. Data selection and feature extraction were performed in the same manner as our previous research, with this first iteration of the lifecycle only including features from the 25 EEG sensors common to all patient data runs (Bowman & Jololian, 2022 Oct). This limitation was decided on to gain familiarity with the data and functionality of various MATLAB toolboxes. Generated software artifacts include the MATLAB scripts for importing EEG and MEG data into MATLAB, scripts to automate feature extraction, and formatted files for importing into WEKA for machine learning model construction and testing. The primary artifact generated in this Phase is shown in Table 1 as it was the artifact most relevant to the domain expert's interests.

Table 1: Confusion matrix from each model; EEG features only, standard 10-fold cross validation

Algorithm	f-measure	rightMesTemp	leftMesTemp	Actual
BayesNet	0.818	5	1	rightMesTemp
		1	4	leftMesTemp
NaiveBayes	1.000	6	0	rightMesTemp
		0	5	leftMesTemp
NaiveBayesUpdateable	1.000	6	0	rightMesTemp

		0	5	leftMesTemp
Logistic Regression	1.000	6	0	rightMesTemp
		0	5	leftMesTemp
SGD	1.000	6	0	rightMesTemp
		0	5	leftMesTemp
Multilayer Perceptron	1.000	6	0	rightMesTemp
		0	5	leftMesTemp
SimpleLogistic	0.818	5	1	rightMesTemp
		1	4	leftMesTemp
SMO (SVM)	0.909	5	1	rightMesTemp
		0	5	leftMesTemp
DecisionStump	0.818	5	1	rightMesTemp
		1	4	leftMesTemp
J48	0.723	5	1	rightMesTemp
		2	3	leftMesTemp
LMT (log tree)	0.818	5	1	rightMesTemp
		1	4	leftMesTemp
Random Forest	0.818	5	1	rightMesTemp
		1	4	leftMesTemp
Random Tree	0.636	4	2	rightMesTemp
		2	3	leftMesTemp

Advancing into Phase 3, we met with the domain expert to discuss these results, some direct observations from reviewing the data files, and some recent journal papers we had discovered after a more extensive review of the literature. It was concluded that these results showed promise for the capability of machine learning in this problem domain. More specifically, the results from this data-driven iteration were incorporated into the domain expert's preconceived hypothesis, slightly altering it based on the degree of success. Thus, the goals of this first iteration were achieved, but the goal of the overall project was, of course, not yet reached. All artifacts were maintained as we cycled back to begin a new iteration through our adaptive lifecycle.

For this next iteration, we chose the data-driven branch to further explore the problem domain using machine learning methodology. We further expanded our exploration to include both EEG and MEG sensors, substantially increasing our feature set. The scripting artifacts and processes developed during the first iteration were altered to account for these changes in hypothesis for the current iteration. Employing the exploration of parallel hypotheses included in our published machine learning framework, we decided upon two complementary hypotheses informed by the problem domain: Localization to the right temporal lobe and localization to the left temporal lobe versus extra-temporal epileptic onset zones with the added parameter of using different feature sets for each hypothesis. Processing during Phase 2 then produced the results seen in Tables 5 and 6. Note that "NaN" denotes a result that is not a number due to division by zero in the calculation of the weighted f-measure for that model.

Table 2: Right temporal vs. bilateral extra-temporal with EEG only, standard 10-fold cross validation

Algorithm	f-measure	rightMesTemp	leftMesTemp	Actual
BayesNet	0.491	0	6	rightMesTemp
		4	11	Extra-temporal
NaiveBayes	0.681	4	2	rightMesTemp
		5	10	Extra-temporal
NaiveBayesMultinomial	?	0	6	rightMesTemp
		0	15	Extra-temporal
Logistic Regression	0.524	1	5	rightMesTemp
		5	10	Extra-temporal

SGD	0.619	2	4	rightMesTemp
		4	11	Extra-temporal
Multilayer Perceptron	0.626	1	5	rightMesTemp
		2	13	Extra-temporal
SimpleLogistic	0.546	0	6	rightMesTemp
		2	13	Extra-temporal
SMO (SVM)	0695	2	4	rightMesTemp
		2	13	Extra-temporal
DecisionStump	0.524	1	5	rightMesTemp
		5	10	Extra-temporal
J48	0.674	3	3	rightMesTemp
		4	11	Extra-temporal
LMT (log tree)	0.546	0	6	rightMesTemp
		2	13	Extra-temporal
Random Forest	0.593	1	5	rightMesTemp
		3	12	Extra-temporal
Random Tree	0.449	1	5	rightMesTemp
		7	8	Extra-temporal

Table 3: Right temporal vs. bilateral extra-temporal with EEG and MEG, standard 10-fold cross validation

Algorithm	f-measure	rightMesTemp	Extra-temporal	Actual
BayesNet	0.657	3	3	rightMesTemp
		4	10	Extra-temporal
NaiveBayes	0.300	3	3	rightMesTemp
		11	3	Extra-temporal
NaiveBayesMultinomial	NaN	0	6	rightMesTemp
		0	14	Extra-temporal

Logistic Regression	0.657	3	3	rightMesTemp
		4	10	Extra-temporal
SGD	0.657	3	3	rightMesTemp
		4	10	Extra-temporal
Multilayer Perceptron	0.600	2	4	rightMesTemp
		4	10	Extra-temporal
SimpleLogistic	0.600	2	4	rightMesTemp
		4	10	Extra-temporal
SMO (SVM)	0613	3	3	rightMesTemp
		5	9	Extra-temporal
DecisionStump	0.567	4	2	rightMesTemp
		7	7	Extra-temporal
J48	0.461	1	5	rightMesTemp
		6	8	Extra-temporal
LMT (log tree)	0.600	2	4	rightMesTemp
		4	10	Extra-temporal
Random Forest	0.609	1	5	rightMesTemp
		2	12	Extra-temporal
Random Tree	0.425	2	4	rightMesTemp
		8	6	Extra-temporal

Upon review with the domain expert during Phase 3 of the current iteration, it was decided that results remain promising, although further exploration was called for with different feature sets and varying research questions. At this stage, data-driven development was generating results which were guiding refinement of the domain expert's hypothesis. Thus, artifacts were maintained as we returned to Phase 1 and began a new iteration through the lifecycle.

Table 4: Left temporal vs. bilateral extra-temporal with EEG only, standard 10-fold cross validation

Algorithm	f-measure	leftMesTemp	Extra-temporal	Actual
BayesNet	0.583	0	5	leftMesTemp
		3	13	Extra-temporal
NaiveBayes	0.861	4	1	leftMesTemp
		2	14	Extra-temporal
NaiveBayesMultinomial	NaN	0	5	leftMesTemp
		0	16	Extra-temporal
Logistic Regression	**0.910**	**5**	**0**	**leftMesTemp**
		2	**14**	**Extra-temporal**
SGD	**0.954**	**5**	**0**	**leftMesTemp**
		1	**15**	**Extra-temporal**
Multilayer Perceptron	**0.954**	**5**	**0**	**leftMesTemp**
		1	**15**	**Extra-temporal**
SimpleLogistic	0.769	3	2	leftMesTemp
		3	13	Extra-temporal
SMO (SVM)	0.810	3	2	leftMesTemp
		2	14	Extra-temporal
DecisionStump	0.676	2	3	leftMesTemp
		4	12	Extra-temporal
J48	0.457	0	5	leftMesTemp
		7	9	Extra-temporal
LMT (log tree)	0.769	3	2	leftMesTemp
		3	13	Extra-temporal
Random Forest	0.791	2	3	leftMesTemp
		1	15	Extra-temporal
Random Tree	0.729	3	2	leftMesTemp
		4	12	Extra-temporal

During this next iteration, we again chose the data-driven branch in Phase 1 and revisited our original research hypothesis of mesial temporal lobe lateralization with a change in our application of the machine learning methodology: Feature subsets. The Execution Phase then called for adjustments to previous artifacts containing labeled feature sets (.arff files for processing in WEKA) instead of changes to scripting. Feedback from the domain expert resulted in changing the results artifacts we present in our meetings to the format seen in Table 8. A development environment designed with our adaptive spiral model in mind could also include the next generation of computer aided software engineering (CASE) tools to support such features such as version control and artifact formatting (Gane, 1988; Kuhn, 1989). This would take some of the organizational workload off of the developer by automating such processes.

Table 5: Right vs Left mesial TLE, standard 10-fold cross validation using the second data file

Algorithm	Beta & low gamma EEG (1st run)	Theta EEG	Theta, Beta, Gamma EEG	EEG&MEG
BayesNet	0.818	0.494	0.494	**0.908**
NaiveBayes	**1.000**	0.636	0.617	0.261
NaiveBayesMultinomial	**1.000**	0.636	0.617	0.291
Logistic Regression	**1.000**	0.617	0.617	0.818
SGD	**1.000**	0.617	0.538	0.545
Multilayer Perceptron	**1.000**	0.617	0.538	0.545

SimpleLogistic	**0.818**	0.425	0.636	0.727
SMO (SVM)	**0.909**	0.353	0.273	0.545
DecisionStump	**0.818**	0.723	0.723	0.727
J48	**0.723**	0.545	0.545	0.545
LMT (log tree)	**0.818**	0.425	0.636	0.727
Random Forest	0.818	0.455	0.636	**0.908**
Random Tree	0.636	0.273	0.617	**0.727**

The Evaluation Phase of this iteration saw our conclusion that clearly some feature subsets were more informative than others, leading to the need to test whether this phenomenon was also true for epilepsy in other regions of the brain.

This idea was tested during the next iteration, again choosing the data-driven branch and employing the machine learning framework for parallelization and efficient exploration. Tables 6 and 7 show the results produced in this iteration. During the Evaluation Phase, these results were deemed encouraging and interesting for further exploration. Specific results showing poor f-measure using only power spectral features in the Delta range (0.1 – 3.5 Hz) in combination with previous hypothesis-driven research led the domain expert to reject the hypothesis that neural activity in this frequency range was relevant to the problem of epileptic focus localization (Berger, 1929). Thus, we encountered an example where the combination of hypothesis- and data-driven approaches resulted in greater refinement of knowledge than either one on their own.

Development returned to the Branching Phase of a new iteration.

Table 6: Right vs Left Frontal, standard 5-fold cross validation

Algorithm	Delta EEG	Theta EEG	Beta and low gamma EEG	Delta, theta, beta, low gamma EEG	Delta, theta, beta, low gamma MEG	EEG&MEG
BayesNet	NaN	NaN	0.556	0.476	0.778	0.778
NaiveBayes	0.556	0.410	0.738	NaN	0.778	0.778
NaiveBayesMultinomial	NaN	NaN	0.667	NaN	NaN	NaN
Logistic Regression	0.646	0.556	**1.000**	0.778	**0.882**	**0.882**
SGD	0.646	0.476	**1.000**	0.738	0.738	0.738
Multilayer Perceptron	0.556	0.556	**1.000**	0.882	**0.882**	**0.882**
SimpleLogistic	0.459	0.567	0.556	0.646	0.646	0.646
SMO (SVM)	0.410	NaN	**0.892**	**0.882**	**0.882**	**0.882**
DecisionStump	0.444	0.459	0.444	0.459	0.556	0.556
J48	0.459	0.459	0.783	0.778	Na N	NaN
LMT (log tree)	0.459	0.567	0.556	0.646	0.646	0.646
Random Forest	0.410	0.476	0.459	0.556	0.646	0.646
Random Tree	0.410	0.410	0.556	0.778	0.410	0.410

Table 7: Right vs Left Frontal, standard 5-fold cross validation

Algorithm	Theta MEG	Beta MEG	low gamma MEG	Beta, low gamma MEG	theta, beta, low gamma MEG
BayesNet	0.646	0.778	0.778	0.778	0.778
NaiveBayes	0.778	**0.882**	0.738	**0.882**	0.778
NaiveBayesMultinomial	NaN	NaN	NaN	NaN	NaN
Logistic Regression	NaN	**0.882**	**0.882**	**0.882**	0.778
SGD	NaN	0.778	**0.882**	**0.882**	0.738
Multilayer Perceptron	NaN	0.738	**0.882**	**0.882**	0.738
SimpleLogistic	0.778	0.738	**0.882**	**0.882**	0.778
SMO (SVM)	NaN	**0.882**	**0.882**	**0.882**	0.738
DecisionStump	0.675	0.778	**0.882**	0.778	0.675
J48	NaN	NaN	NaN	NaN	NaN
LMT (log tree)	0.778	0.738	**0.882**	**0.882**	0.778
Random Forest	0.738	0.476	0.738	0.778	0.778
Random Tree	0.646	0.646	0.778	0.778	0.567

In this new iteration, we adjusted our hypothesis to once again explore mesial temporal lateralization as well as another concept in parallel. Proceeding again into the data-driven branch of Phase 2 in our lifecycle, we split development into two tracks, one for refined application of machine learning while another explored directed transfer functions, another data-driven methodology. Previous literature had shown promising results with small sample sizes, thus we were interested in potentially replicating some of those results and exploring a different avenue of data-driven research (Dai & He, 2011).

Table 12 shows our results from the machine learning research track while Figure 7 shows some results from the DTF research track, generated from data of one patient with left mesial temporal lobe epilepsy. Exploration of this second research track was thereby facilitated due to reuse of data and engineering artifacts, primarily the scripting used to import and format the MEG data.

Table 8: Left mesial temporal vs right mesial temporal, 10-fold cross validation

Algorithm	Theta MEG	Theta EEG	Beta, Gamma EEG	Theta, Beta, Gamma EEG	EEG&MEG
BayesNet	0.646	0.494	0.818	0.636	**0.908**
NaiveBayes	0.778	0.636	**1.000**	**0.909**	0.261
NaiveBayesMultinomial	NaN	0.636	0.091	0.091	0.291
Logistic Regression	NaN	0.617	**1.000**	0.815	0.818
SGD	NaN	0.617	**1.000**	**0.909**	0.545
Multilayer Perceptron	NaN	0.617	**1.000**	**1.000**	0.545
SimpleLogistic	0.778	0.425	0.818	0.815	0.727
SMO (SVM)	NaN	0.353	**0.909**	**0.909**	0.545
DecisionStump	0.675	0.723	0.818	0.818	0.727
J48	NaN	0.545	0.723	0.538	0.545
LMT (log tree)	0.778	0.425	0.818	0.727	0.727
Random Forest	0.738	0.455	0.818	0.723	**0.908**
Random Tree	0.675	0.273	0.636	**0.909**	0.727

Figure 7 below shows the results from a single data file while considering only the Theta frequency band, commonly defined as 3 – 8 Hz (Berger, 1929).

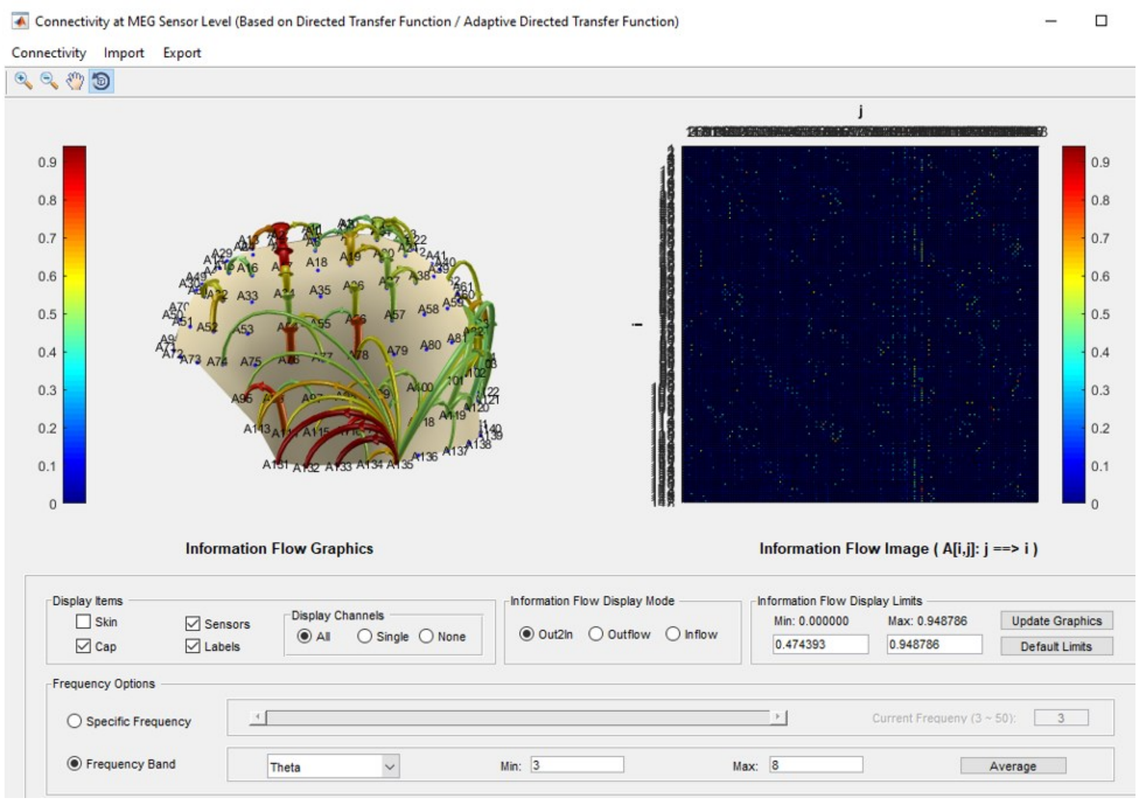


Figure 7: DTF results from one data run of a patient with left mesial temporal lobe epilepsy

Discussion

From an engineering standpoint, our adapted spiral model provides the flexibility for the developer(s) to seamlessly switch between hypothesis- and data-driven development as needed. This organizational structure to development then allows for developers to benefit from the advantages of each approach. With our previously published machine learning framework embedded into the data-driven branch of this model, the developer gains greater access to the power of the machine learning paradigm. The parallel construction and testing of machine learning models with various feature sets will scale with improvements in hardware. Increased parallel performance will further enable feasible exploration of the solution space, including different aspects of machine learning methodology such as different preprocessing techniques and fine-tuning hyper-parameters. Parallel exploration of research questions will scale with the shift in the mindset of the developer away from the traditional approach of only considering a single hypothesis at a time. Furthermore, the reuse of data and feature sets implicit within our model allows for this scaling to be performed efficiently. When considered together, these scaling aspects of our model allow for research projects to achieve a greater scope (in depth or breadth) while accelerating a more extensive application of machine learning methodologies. Future research can also incorporate other aspects into this adaptive spiral model, such as further developing the intelligent features in our machine learning framework or the modifications based on management theory later published as the WinWin spiral model (Boehm et al, 1998).

From a medical standpoint, the results shown here reinforce some previous studies by other research groups while also exhibiting some interesting phenomena in other aspects dependent on the specific hypothesis. For example, the results shown in Tables 2 and 4 reinforce the apparent disparity in discriminating left versus right, in the frontal lobe in these results and mirrored in the temporal lobes in our previous research (Bowman & Jololian, 2022). Results showing models achieving much higher f-measure when identifying left frontal lobe epilepsy against extra-temporal examples versus those same models trained with the same feature set attempting to identify right frontal lobe epilepsy. While our results are limited by a small sample size, we believe this disparity warrants further investigation to elucidate possible neurophysiological differences between lobes. These differences in functional characteristics may then explain observed phenomena and thereby influence clinical decisions based on anatomical location, supplementing functional mapping typically performed prior to epilepsy surgery (Knowlton, 2006; Pataraia et al, 2004). Such variable functional characteristics between sides of the same lobe may also explain and contribute to the variability of results seen in previous studies exploring cortical connectivity through transfer entropy and graph theoretic metrics (Basu et al, 2015; Wu et al, 2018).

Conclusions

In this research, we presented an adaptation of the spiral model to incorporate both hypothesis- and data-driven approaches. Combined with our previously published machine learning framework, this lifecycle model now provides greater flexibility for the developer in designing solutions appropriate for the problem at hand while allowing for greater adoption of the machine learning paradigm. Through parallelism and reuse of data and feature subsets, this lifecycle leads the developer to efficiently generate results of machine learning applications, providing a path to better models through a wider search through the solution space. The walkthrough provided in this work shows that applying our model to a complex medical problem yields better results than previous published attempts, even while employing a suboptimal implementation with the development tools immediately available. Discussion in this research has shown how the benefits of this adapted model can be further amplified if a development environment was designed with this model in mind. Combined with other supporting features, such a development environment would also provide greater accessibility to the power of the machine learning paradigm for engineering and domain experts alike.

Acknowledgements

This work was supported by an institutional grant from the University of Alabama at Birmingham. We would also like to acknowledge the contributions of Dr. Ismail Mohamed and Jeff Killen in data collection, management, and selection.

References

Aoe, J., Fukuma, R., Yanagisawa, T., Harada, T., Tanaka, M., Kobayashi, M., ... & Kishima, H. (2019). Automatic diagnosis of neurological diseases using MEG signals with a deep neural network. *Scientific reports*, *9*(1), 1-9.

Basu, I., Kudela, P., Korzeniewska, A., Franaszczuk, P. J., & Anderson, W. S. (2015). A study of the dynamics of seizure propagation across micro domains in the vicinity of the seizure onset zone. *Journal of neural engineering*, *12*(4), 046016.

Berger H. Über das elektroenkephalogramm des menschen. Archiv für Psychiatrie und Nervenkrankheiten. 1929;87:527

Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, *21*(5), 61-72.

Boehm, B., Egyed, A., Kwan, J., Port, D., Shah, A., & Madachy, R. (1998). Using the WinWin spiral model: a case study. *Computer*, *31*(7), 33-44.

Bowman, A. D., & Jololian, L. (2021). A conceptual framework for an introductory machine learning course. *Journal of Computing Sciences in Colleges*, *37*(1), 78-83.

Bowman, A. D., Prabhakar, S. P., & Jololian, L. (2022, March). A Framework for an Automated Development Environment to Support the Data-driven Machine Learning Paradigm. In *SoutheastCon 2022* (pp. 329-331). IEEE.

Bowman, A. D., & Jololian, L. (2022, Oct). *A Case Study in a Machine Learning Framework Applied to Epilepsy Localization* [Manuscript submitted for publication]. Department of Electrical and Computer Engineering, University of Alabama at Birmingham.

Coolen, T., Dumitrescu, A. M., Bourguignon, M., Wens, V., Urbain, C., & De Tiège, X. (2018). Presurgical electromagnetic functional brain mapping in refractory focal epilepsy. *Zeitschrift für Epileptologie*, *31*(3), 203-212.

Dai, Y., & He, B. (2011, May). MEG-based brain functional connectivity analysis using eConnectome. In *2011 8th International Symposium on Noninvasive Functional Source Imaging of the Brain and Heart and the 2011 8th International Conference on Bioelectromagnetism* (pp. 9-11). IEEE.

Dai, Y., Zhang, W., Dickens, D. L., & He, B. (2012). Source connectivity analysis from MEG and its application to epilepsy source localization. *Brain topography*, *25*(2), 157-166.

Elisevich, K., Shukla, N., Moran, J. E., Smith, B., Schultz, L., Mason, K., ... & Bowyer, S. M. (2011). An assessment of MEG coherence imaging in the study of temporal lobe epilepsy. *Epilepsia*, *52*(6), 1110-1119.

England, M. J., Liverman, C. T., Schultz, A. M., & Strawbridge, L. M. (2012). Epilepsy across the spectrum: Promoting health and understanding: A summary of the Institute of Medicine report. *Epilepsy & Behavior*, *25*(2), 266-276.

Englot, D. J., Nagarajan, S. S., Imber, B. S., Raygor, K. P., Honma, S. M., Mizuiri, D., ... & Chang, E. F. (2015). Epileptogenic zone localization using magnetoencephalography predicts seizure freedom in epilepsy surgery. *Epilepsia*, *56*(6), 949-958.

Gane, C. (1988). *Computer-aided software engineering: the methodologies, the products, the future*. Prentice-Hall, Inc..

Guo, J., Yang, K., Liu, H., Yin, C., Xiang, J., Li, H., ... & Gao, Y. (2018). A stacked sparse autoencoder-based detector for automatic identification of neuromagnetic high frequency oscillations in epilepsy. *IEEE transactions on medical imaging*, *37*(11), 2474-2482.

Haneef, Z., Lenartowicz, A., Yeh, H. J., Levin, H. S., Engel Jr, J., & Stern, J. M. (2014). Functional connectivity of hippocampal networks in temporal lobe epilepsy. *Epilepsia*, *55*(1), 137-145.

He, B., Dai, Y., Astolfi, L., Babiloni, F., Yuan, H., & Yang, L. (2011). eConnectome: A MATLAB toolbox for mapping and imaging of brain functional connectivity. *Journal of neuroscience methods*, *195*(2), 261-269.

Kamiński, M., Ding, M., Truccolo, W. A., & Bressler, S. L. (2001). Evaluating causal relations in neural systems: Granger causality, directed transfer function and statistical assessment of significance. *Biological cybernetics*, *85*(2), 145-157.

Knowlton, R. C. (2006). The role of FDG-PET, ictal SPECT, and MEG in the epilepsy surgery evaluation. *Epilepsy & Behavior*, *8*(1), 91-101.

Kugiumtzis, D. (2001). On the reliability of the surrogate data test for nonlinearity in the analysis of noisy time series. *International Journal of Bifurcation and Chaos*, *11*(07), 1881-1896.

Kuhn, D. L. (1989). *Selecting and effectively using a computer aided software engineering tool* (No. WSRC-RP-89-483; CONF-891192-7). Westinghouse Savannah River Co., Aiken, SC (United States).

MATLAB. (2018). *9.4.0.813654 (R2018a)*. Natick, Massachusetts: The MathWorks Inc.

Pataraia, E., Simos, P. G., Castillo, E. M., Billingsley, R. L., Sarkari, S., Wheless, J. W., ... & Papanicolaou, A. C. (2004). Does magnetoencephalography add to scalp video-EEG as a diagnostic tool in epilepsy surgery?. *Neurology*, *62*(6), 943-948.

Soriano, M. C., Niso, G., Clements, J., Ortín, S., Carrasco, S., Gudín, M., & Pereda, E. (2017). Automated detection of epileptic biomarkers in resting-state interictal MEG data. *Frontiers in neuroinformatics*, *11*, 43.

Spencer, S. S. (2002). Neural networks in human epilepsy: evidence of and implications for treatment. *Epilepsia*, *43*(3), 219-227.

Tadel, F., Baillet, S., Mosher, J. C., Pantazis, D., & Leahy, R. M. (2011). Brainstorm: a user-friendly application for MEG/EEG analysis. *Computational intelligence and neuroscience*, *2011*.

Theiler, J., Galdrikian, B., Longtin, A., Eubank, S., & Farmer, J. D. (1991). *Testing for nonlinearity in time series: the method of surrogate data* (No. LA-UR-91-3343; CONF-9108181-1). Los Alamos National Lab., NM (United States).

Witten, I. H., Frank, E., Trigg, L. E., Hall, M. A., Holmes, G., & Cunningham, S. J. (1999). Weka: Practical machine learning tools and techniques with Java implementations.

Ursino, M., Ricci, G., & Magosso, E. (2020). Transfer entropy as a measure of brain connectivity: A critical analysis with the help of neural mass models. *Frontiers in computational neuroscience*, *14*, 45.

Wang, D., Ram, P., Weidele, D. K. I., Liu, S., Muller, M., Weisz, J. D., ... & Amini, L. (2020, March). Autoai: Automating the end-to-end ai lifecycle with humans-in-the-loop. In *Proceedings of the 25th International Conference on Intelligent User Interfaces Companion* (pp. 77-78).

Witten, I. H., Frank, E., Trigg, L. E., Hall, M. A., Holmes, G., & Cunningham, S. J. (1999). Weka: Practical machine learning tools and techniques with Java implementations.

Wu, T., Chen, D., Chen, Q., Zhang, R., Zhang, W., Li, Y., ... & Zhang, J. (2018). Automatic lateralization of temporal lobe epilepsy based on MEG network features using support vector machines. *Complexity*, *2018*.

FUTURE APPLICATIONS

Overview

While machine learning has gained traction in a wide variety of domains in recent years, one area of particular interest for future development is the smart city. In this section we will present a simple example from day-to-day life in a smart city, then describe how machine learning is involved in different aspects of that example. We will discuss ongoing research in related domains to illustrate the potential for machine learning methodology to provide solutions to everyday problems within the context of a smart city. We will then include a paper we intend to submit for publication detailing how our novel lifecycle can facilitate the use of machine learning within the context of a smart city.

Smart City Background

The development of a smart city within the urban planning context includes far more than only the advance planning of utilities for easier, stable expansion of residential and commercial districts. Analysis of information gathered from both other cities and the development of the current city can then be used to inform plans for future infrastructure or redevelopment. Surveillance showing a substantial pocket of younger, more tech-oriented families settling into a certain area of the city, stimulating growth in that direction can be used to justify focusing on improving the utilities, widening the roads, and making sure to designate an appropriate amount of land area for businesses and schools. Doing so will lay the groundwork to support the population growth anticipated in that region, making the incorporation of surveillance data into urban policy a “smart” process. On the other hand, if surveillance shows little to no influx of young[er] people into an area, that area may be considered for redevelopment. Older, defunct businesses may be removed to make way for newer businesses, including retirement homes/communities to accommodate the aging population of the region. In this way, the smart policy can anticipate the needs of the smart city on a region-specific basis, determined wherever there are distinct trends in the population.

Shifting to the transportation context, the gradual accommodation of today's tech-related needs can be seen in major airports across the country. Where at first airport seating began providing additional power outlets as passengers needed a place to charge their laptop, more and more airports are now also including designated charging stations to charge all manner of personal electronic devices from laptops to phones. The rapid growth of the smartphone industry and the population's increased adoption and use of their devices, especially while traveling, necessitated airports to better support those devices. This relatively small-scale example shows the extent of which today's population is increasingly connected digitally, lending itself to the "digital city" label where businesses from all manner of industries also cater to this drive from customers. Advertisements of "free Wi-Fi" at hotels, airports, and restaurants are still only early, small scale examples of what a fully connected, digital city could be. Integrated systems within a city's infrastructure can aid in easing traffic congestion, possibly avoiding it entirely. A very basic example is a sensor at an intersection to detect when a vehicle is in the turn lane. If so, the system governing the traffic lights incorporates that information into its sequence of which lights to turn green, ensuring traffic continues to flow through the intersection. In a more realized intelligent city, this information will be gathered, refined, and processed on a larger scale by a centralized intelligent system. By coordinating the timing of green lights through neighboring intersections, the system can then ensure that traffic flows as smoothly as possible throughout the city, not limited to only one intersection. Thus, a smart city can be defined as one which uses the knowledge gained from surveillance of the population combined with the latest technology to better anticipate and provide the needs of that population.

The capabilities of a smart city can also be an incredible boon to the economy of the private sector. In a smart home equipped with such products as a smart refrigerator, thermostat, lighting, and other appliances, the needs of the citizen at home can be met in an automated fashion with little to no input from the person. The smart refrigerator can be set up with the option to automatically detect when the person is running low on certain grocery items deemed a high priority to keep stocked. The networked refrigerator sends the owner a message with a list of groceries to confirm which ones to restock, then connects to all grocery stores on the city-wide network provided by the smart city. Further customization options could include the owner selecting a list of brands and sizes they prefer for each item. The intelligent system embedded within the smart refrigerator could then query every smart grocery store connected to the city's network for the current prices of each selected item from the preferred brands and sizes. Comparing different prices and taking into account any pertinent delivery fees, the refrigerator could then *optimize* the grocery order and send the order(s) to the grocery store(s) at a preset time. Going a step further, the smart home could learn to predict ahead of time when the person will need the next gallon of milk based on how quickly they drink milk in the past. Taking that another step further, it uses those predictions when calculating optimal grocery lists. It could be that ordering the 5 items today will cost $10, buying these items from a certain set of stores. However, the system predicts the person will need another gallon of milk tomorrow, which costs $2 plus a convenience fee of $1. Instead of placing the $10 order today and $3 order tomorrow, it sees that it is able to wait an extra day and save the $1 convenience fee on the milk by adding it on to an order. In this way, the owner's smart refrigerator offers both the convenience of automated grocery ordering and stocking and potential monetary savings from optimizing each individual order through comparison shopping and predicting the owner's needs. This is aside from the time savings of having the groceries delivered to the owner's home address.

From the grocery store's perspective, smart homes equipped with smart refrigerators could mean a boon to their business volume. Proceeding with the aforementioned example, the store's system receives an order over the network with a delivery address included. The system can then calculate the time required for a delivery vehicle to reach that address from the store as well as estimate when a self-driving delivery vehicle will be next available. The system provides that information to an employee to gather the groceries and place them within the delivery vehicle. Said vehicle navigates to the provided address using infrastructure built into the roads of the smart city. Once the customer has retrieved their groceries (verified by internal sensors within the grocery compartment of the vehicle) and closes the compartment, the vehicle either returns to the store or proceeds directly to the next address in the queue to deliver the next batch of groceries in a different compartment. The appropriate grocery compartment on the vehicle can automatically open once the customer approaches the vehicle and their identity confirmed by such means as a preset password or facial recognition (their image compared to the image in the city's database of citizens). Orders can be placed into the same delivery vehicle based on optimal driving paths i.e. if two delivery addresses are in the same neighborhood or nearby, it is more efficient to have the same vehicle delivery to both address without returning to the store, assuming the intended delivery time selected by the customers allow for it. In this way, the potential number of vehicles on the roads can be reduced as there are fewer customers driving to the stores and fewer delivery vehicles. This results in the street infrastructure of the smart city adequately providing the needed space to handle traffic for a greater length of time into the future before needing to be widened. This ensures the available infrastructure does not strangle any economic growth for as long as possible. Thus, a smart city can provide the infrastructure to be the backbone which smart homes and smart businesses rely on to energize a stable and growing economy.

As a commercial example within a smart city, a smart home equipped with network-enabled devices can anticipate future needs of its residents and facilitate transactions with businesses to fulfill those needs. A smart refrigerator may be set up with the option to automatically detect when supply on certain, high priority grocery items is running low. Upon detection, the refrigerator could connect to all grocery stores on the city-wide network provided by the smart city. After gathering the pricing and supply information from each store, the refrigerator would present this list to the resident to facilitate the purchase of the desired groceries. From the grocery store's perspective, smart homes equipped with such smart refrigerators could mean a boon to their business volume. Proceeding with this example, the store's system receives an order over the network with a delivery address included. The system can then calculate the time required for a delivery vehicle to reach that address from the store as well as estimate when a self-driving delivery vehicle will be next available. The system provides that information to an employee to gather the groceries and place them within the delivery vehicle. Said vehicle navigates to the provided address using infrastructure built into the roads of the smart city. Once the customer has retrieved their groceries (verified by internal sensors within the grocery compartment of the vehicle) and closes the compartment, the vehicle either returns to the store or proceeds directly to the next address in the queue to deliver the next batch of groceries in a different compartment. The appropriate grocery compartment on the vehicle can automatically open once the customer approaches the vehicle and their identity confirmed by such means as a preset password or facial recognition (their image compared to the image in the city's database of citizens). Orders can be placed into the same delivery vehicle based on optimal driving paths i.e. if two delivery addresses are in the same neighborhood or nearby, it is more efficient to have the same vehicle delivery to both address without returning to the store, assuming the intended delivery time selected by the customers allow for it. In this way, the potential number of vehicles on the roads can be reduced as there are fewer customers driving to the stores and fewer delivery vehicles. This results in the street infrastructure of the smart city adequately providing the needed space to handle traffic for a greater length of time into the future before needing to be widened. This ensures the available infrastructure does not strangle any economic growth for as long as possible.

Thus, a smart city can provide the infrastructure to be the backbone which smart homes and smart businesses rely on to energize a stable and growing economy.

Within this example we see machine learning applied in a wide variety of ways, some more apparent than others. The more prominent applications include computer vision in the self-driving delivery vehicles and facial recognition for customer identification. Perhaps less obvious are the applications that function behind the scenes, such as machine learning models working to detect fraudulent transactions, other models optimizing the traffic flow through the city among other infrastructure, or the natural language processing utilized if the customer uses a search feature to add other grocery items to the shopping list since they already plan to order some essentials. The variety of applications shown within this one context offers a glimpse into the power offered by the machine learning paradigm. However, this does not mean data-driven methodologies like machine learning are the be-all-end-all for all research and development. For this next section we take a step back from smart cities to discuss research methodologies in broad strokes.

An Integrated Intelligent System for the Automatic Detection of Patients at High Risk for a Cardiovascular Event

ANTHONY D. BOWMAN, LEON JOLOLIAN

Submitted to SoutheastCon 2023

Format adapted for dissertation

Abstract

With the recent and growing emergence of the machine learning paradigm as a data-driven approach to problem solving, researchers and software developers alike face the challenge of how to apply those techniques to increasingly complex problems. In developing solutions, researchers often focus their efforts on a single hypothesis, choosing only one or two machine learning algorithms trained with a static feature set. This unnecessarily narrow focus to development represents the application of a data-driven approach from a hypothesis-driven mindset, imposing a bottleneck on both the potential results and the overall scope of the project. This limitation is reinforced by the limitations of current development environments. To address this, we have previously published a machine learning framework to release both the bottleneck on achievable results and scope. In this paper, we present an intelligent system capable of driving retrospective research and prospective health surveillance when designed with our framework in mind. By design, this system would facilitate application of the machine learning paradigm in research while allowing for the deployment of solutions into a healthcare environment. When deployed within a smart city, we discuss how the benefits of this system would be amplified to entire urban population beyond a single healthcare organization.

Keywords—intelligent system, machine learning, software development, smart city

Introduction

Advances in machine learning techniques have led to substantial progress in a variety of domains. Among these is the widespread application of machine learning methodology to complex problems in the medical domain. Researchers have now employed machine learning for data-driven investigation and development of solutions to problems ranging from diagnosis of medical imaging and cancer to identification of neurological disorders such as epilepsy (Abdar & Makarenkov, 2019; Wu et al, 2018; Zhang, Wang, Liu, & Yang, 2016). While previous research has seen some incredible results, often the application of machine learning is unnecessary limited by selection of only one or two classifiers, often without justification. We have previously published a framework which facilitates the adoption of the machine learning paradigm, removing such limitations which restricted the scope and success of applying machine learning (Bowman & Jololian, 2021; Bowman, Prabhakar, & Jololian, 2022). While the volume of medical research employing machine learning continues to increase, there is a need to develop methods of translating that research into practical tools to be deployed in the clinical setting. With the widespread prevalence of electronic medical record (EMR) systems, an intelligent system can bridge the divide between the research and clinical settings. In this research, we describe such an intelligent system which, designed and trained with our framework in mind, provides an integral tool to conduct data-driven research and reduce the effort needed to deploy developed solutions into the clinical setting.

In the United States today, heart disease is the leading cause of death in the adult population (Heron, 2018). A system for identifying individuals at high risk for a cardiovascular event such as a myocardial infarction in the near future continues to elude researchers (Eagle et al, 2010). This research aims to fill this gap in the healthcare industry, providing a flexible software solution that can both adapt to changes in focus, data, or its environment as well as scale up or down to meet the needs of its target population. While the discussion in this paper is primarily focused on heart disease due to its prevalence and mortality, the intelligent system detailed here can also function in a similar capacity to serve the needs of clinicians for identifying a multitude of diseases or disorders.

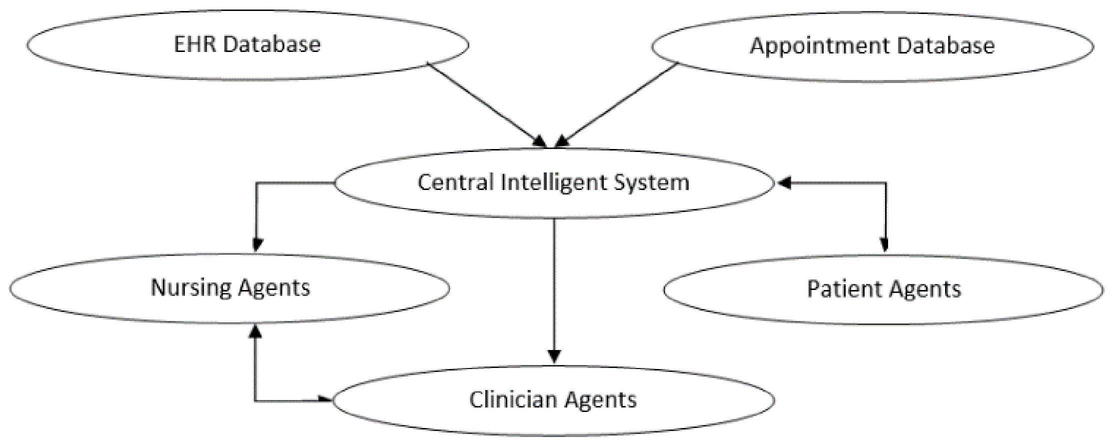


Fig 1. High level agent diagram

Intelligent System Base Design

At its core, the proposed solution is an intelligent system embedded within the EMR system of a medical provider, be it a large medical center or small health clinic. The system gathers feature data from the EMR system and automatically labels them into two classes depending on the presence or absence of a cardiovascular event in that patient's medical history. The system then trains multiple algorithms with the labeled data from a specific time frame, testing each algorithm using cross validation and labeled data from outside the selected time frame. After this initial training and testing period, the system is fully deployed for use in the clinical setting where it immediately classifies using each algorithm each patient encountered in every clinic each day as at risk of a cardiovascular event or not. If so, a message is automatically generated by the system and sent to the charge nurse of the clinic and the clinician seeing the patient that day. This message contains both the classification results from each algorithm and an up-to-date confusion matrix showing the type I and type II error along with overall accuracy/precision. The clinician can then make an informed decision on how to address the prediction by the system. At the end of each day, new patient information regarding both patients with newly added cardiovascular events in their EMRs and brand-new patients with non-cardiovascular diagnoses will be added to the data set for the system to train and test itself. In this way, the system adapts to both changes in data and environment as the demographic s of the local population may change over time, gradually accounted for automatically by the system retraining. This system is also scale-able from a small health clinic to a large medical center as both will likely have more than sufficient historical medical data to train the algorithms. Similar systems using machine learning algorithms have previously been used to great effect for classification of mammogram images to detect potentially pathogenic abnormalities (Zhang, Wang, Liu, & Yang, 2016). However, this proposed system takes a much more proactive approach to surveillance, with a far broader scope not limited to the results of a single test. After successful deployment and use in the clinical setting, the system can then be used as the basis for an even wider scale deployment in the mobile market, using the current algorithm parameters, sensors, and a short questionnaire to gather the user's feature set and calculate a preliminary classification. Frequent updates to the mobile app can be the vector to deliver the most up-to-date algorithm parameters as the intelligent system continues to train and test with new data in the clinical setting.   A similar approach with an ensemble machine learning model has been shown to have acceptably high accuracy and be suitable for use in the telehealth market (Zhang et al., 2017).

The overall design of this intelligent system is allowed to emerge naturally from the process of its creation. Beginning with an objected oriented design process in mind, all potential objects in mind are treated as agents. Each agent has associated actions, a list of other agents it interacts with through those actions, and data storage for information kept internally within the agent. In the case of this intelligent system, there is a centralized processing agent, satellite clinical agents, and satellite patient agents. The clinical agents can be further divided into clinician and nursing agents to distinguish between the information provided to each and the urgency of that information. Because this is an embedded system, there are other key components it indirectly interacts with that may not be explicitly shown in this model, including the systems for gathering certain medical data such as those in radiology. While the proper functioning of this system requires data from those, it retrieves that data from a centralized EMR database which serves as the middleman. Figure 1 shows this general structure with the direction of information flow from each agent.

Smart City Context

Across the globe a mass migration of the human population continues as it has for the last century, away from rural living and into the metropolis of urban life. With this transition, the world's cities face the issues which stem from accommodating an ever-growing populace, now on an even greater scale than ever before in history (Caragliu, Del Bo, & Nijkamp, 2011). These not only include the basic needs of clean water, sanitation and public health, reliable power, well maintained roads and traffic systems but also a growing reliance on reliable broadband internet, information and communication technologies (ICTs), and the flexibility of the city's infrastructure to adapt to the needs of the people. Out of this need grew the concept of the "smart city," a term without a clear definition but invoking a general idea of designing a city capable of not only allowing its people to live but thrive in a technological world. Using the tools available within this smart city context, this project seeks to combat a growing health problem: Cardiovascular events caused by heart disease.

The concept of a smart city has been used and labeled with many different names including intelligent city, knowledge city, sustainable city, digital city, etc. (Cocchia, 2014). In keeping with the ambiguity of natural languages, the selection of which name to use and its meaning depend on the context. If viewed in urban development, a smart city could be one where the government and public agencies adopt policies and strategies to promote and allow for the sustainable and stable growth of the community (Nam & Pardo, 2011). Keeping with the urban planning context, the construction of infrastructure to accommodate not only the current population but the forecasted future population after years of growth could also be considered "smart." This could include wider roads than is currently needed by today's traffic or the strategic placement of additional water treatment plants to serve future housing developments in the suburban areas. Often coupled with the water and sewer infrastructure is the expansion of the power grid to provide a backbone for other utilities such as broadband internet. Some cities have also taken the extra step by building the infrastructure for city-wide broadband internet, considering it an essential public utility alongside water and power (Nam & Pardo, 2011).

On an even broader scale, some nation states have done this with high-speed broadband internet in place throughout their country, providing the ICT basis for rapid technological expansion as their population grows. This forward-thinking approach to water/sewer, power, and internet expansion thus allows for not only residential but industrial growth, particularly in the case of high-speed internet. With technological companies becoming more reliant on access to or delivery of cloud-based computing solutions, the availability of high-speed broadband internet is critical to their success. Pre-existing broadband infrastructure will likely yield substantial dividends in the long term as the tech industry can hit the ground running. This head start reduces the initial time and cost for businesses to move into a location within the city, set up their operations and begin providing their goods and services. The planning for these utilities thus enables jumpstarting the local economy by easing the barrier to entry, potentially reducing the time between when a company establishes its physical presence and when it can begin hiring the human workforce to use those utilities and begin generating revenue. Such advance planning for the expansion of utilities can then be considered the beginning of a smart city.

Smart planning of infrastructure and a fully connected, digital city can also have substantial benefits regarding the health of the population. In a smart city with the required networking infrastructure, the emergency medical services (EMS) can be optimized to ensure assistance is rendered as fast as possible by reducing the latency between first responders reaching the person in need as well as the time it takes to transport that person to the nearest emergency department if needed. This can be done by automatically calculating and selecting the ambulance, fire truck, or police car which can reach the person as fast as possible. Such calculation can be fully informed by the location of the person (gathered by tracing the call or other means), the GPS in the EMS vehicle providing their current location and city-wide surveillance providing up-to-date traffic information between that vehicle and the target person. This system can then also provide the optimal driving path (i.e. driving directions) to that vehicle's onboard computer through the city-wide network, even going as far as altering the traffic lights along their route to further ensure minimal driving time. If the first responders are on foot (i.e. outside their vehicle and inside a strictly pedestrian zone such as a shopping mall), the system can instead provide the most direct walking path based on the schematics of the building(s) as provided to the city by their architect(s). If the first responders signal to the system the person requires further, immediate medical attention, the system can consider every emergency department within a set distance. The system can then calculate the optimal path from the first responders to the nearest emergency department that is able to provide immediate care to that patient. This excludes all emergency departments that may be closer but currently have a high patient load and would not be able to render medical assistance as quickly as the next department. This additional screening of emergency departments would be especially critical in distributing the patient load throughout the medical network in the event of a mass casualty event. With the added information of a patient's critical status (how quickly they need care i.e. their priority as determined by the medical personnel in the field), the system can direct ambulances with the more critical patients to the nearest emergency departments capable of providing the immediate care they need. By directing ambulances with less critical patients to more distant hospitals, the system distributes the patient load and attempts to prevent the death of critical patients because nearby hospital resources were already overwhelmed. After delivery of the patient, the EMS crew can fill out their section of information concerning the day's events and submit it into the smart city's centralized electronic medical record (EHR) system. That information is then immediately available for review by any medical personnel looking at the patient's medical record. The first responders can then signal to the system they are once again available for the next prospective patient.

Expanded Design within Smart Cities

The inclusion of this system within a smart city greater expands its potential deployment options and benefits to the population. With every clinic, hospital, and pharmacy connected to form a city-wide healthcare network, the proposed intelligent system can be embedded within the unified EMR used at every site. This allows the system to access and incorporate data gathered at every site into its ensemble of machine learning algorithms. This also allows all patients to visit any clinic or hospital they choose, whether it be a preference of proximity to their home/work or their preference of physician/healthcare provider. On the reverse side of continually training, the system can also extend its reach out beyond only hospitals to the smaller, more specialized clinics to relay information concerning their patients with appointments each day. The specialists providing care at those clinics can then be automatically informed if their patient has been classified as at high risk for a cardiovascular event within the next set time period. This information may play a more critical role in their diagnosis and/or treatment of seemingly unrelated symptoms, thereby avoiding a misdiagnosis. The information can also be used by pharmacists whenever the patient goes to refill a prescription for a medication contraindicated for someone with heart disease or related illness. The flow of information through the city-wide network can then be viewed as Figure 2 below. Thus, the infrastructure provided by a smart city, namely the healthcare network of clinics, hospitals, and pharmacies all using the same EMR, affords the intelligent system the potential capability of preventing costly misdiagnoses and ill-advised medications for better personalized medicine.

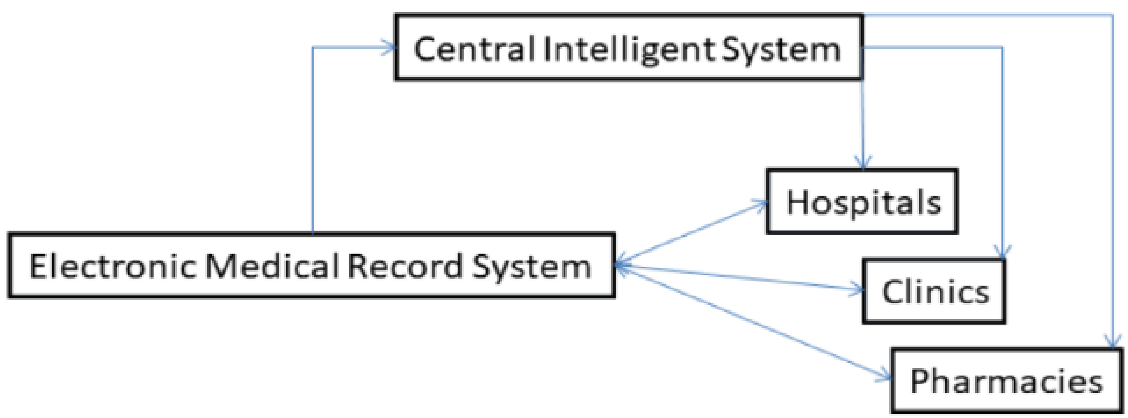


Fig 2. Diagram of integration into a smart city healthcare network

Summary

In this research, we provide the high-level design of an intelligent system designed with our previously published machine learning framework in mind. Employing the processing pipeline envisioned by that framework, this system facilitates the application of machine learning to a broad range of problems through its interaction with an electronic medical record system. Parallel investigations are efficiently performed through the re-use of data and feature sets while broadening the scope of research and providing more optimal solutions through a more thorough search through the solution space. In addition to supporting data-driven medical research, this system would also allow for developed solutions to be deployed into the healthcare setting, facilitating the translation of research into applied work. The incorporation of this intelligent system into the healthcare network of a smart city provides additional benefits to the population by extending its reach beyond the clinical setting.

References

[1] Abdar, M., & Makarenkov, V., "CWV-BANN-SVM ensemble learning classifier for an accurate diagnosis of breast cancer," in *Measurement*, *146*, 557-570, 2019.

[2] T. Wu, D. Chen, C. Qiqi, R. Zhang, W. Zhang, Y. Li, L. Zhang, H. Liu, S. Wan, T. Jiang and J. Zhang, " Automatic lateralization of temporal lobe epilepsy based on MEG network features using support vector machines," *Complexity,* no. Feb, 2018.

[3] Zhang, Y. D., Wang, S. H., Liu, G., & Yang, J.. "Computer-aided diagnosis of abnormal breasts in mammogram images by weighted-type fractional Fourier transform," *Advances in Mechanical Engineering*, *8*(2), 2016.

[4] Bowman, A. D. and L. Jololian, "A conceptual framework for an introductory machine learning course," *Journal of Computing Sciences in Colleges*, Lacey, WA, 2021.

[5] Bowman, A. D., Prabhakar, S. P., & Jololian, L., "A Framework for an Automated Development Environment to Support the Data-driven Machine Learning Paradigm," *SoutheastCon 2022*, pp. 329-331, 2022.

[6] Heron , M. P., "Deaths: leading causes for 2016." *NVSR*, *67*, 1-75, 2018.

[7] Eagle, K. A., Ginsburg, G. S., Musunuru, K., Aird, W. C., Balaban, R. S., Bennett, S. K., ... & Greenland, P., "Identifying patients at high risk of a cardiovascular event in the near future: current status and future directions: report of a national heart, lung, and blood institute working group," *Circulation*, *121*(12), 1447-1454, 2010.

[8] Zhang, J., Lafta, R. L., Tao, X., Li, Y., Chen, F., Luo, Y., & Zhu, X., "Coupling a fast fourier transformation with a machine learning ensemble model to support recommendations for heart disease patients in a telehealth environment," *IEEE Access*, *5*, 10674-10685, 2017.

[9] Caragliu, A., Del Bo, C., & Nijkamp, P., "Smart cities in Europe," *Journal of urban technology*, *18*(2), 65-82, 2011.

[10] Cocchia, A., "Smart and digital city: A systematic literature review," in *Smart city,* pp. 13-43, 2014.

[11] Nam, T., & Pardo, T. A., "Conceptualizing smart city with dimensions of technology, people, and institutions," in Proceedings of the 12th annual international digital government research conference: digital government innovation in challenging times, pp. 282-291, June 2011.

CONCLUSION

In this research, we propose a development environment that allows researchers to leverage those capabilities more fully by shifting not only the tool they use but also their mindset. We also identify several factors that affect the efficacy and productivity of this tool and the solutions it generates. Our proposed environment serves as an intermediate tool, guiding the researcher and making full adoption of the machine learning paradigm throughout the software development process easier. To accomplish this, our framework is defined by a three-layer structure designed for subject domain assessment, data manipulation and feature set exploration. Supported by parallelism, data cleaning and feature engineering, this research provides a conceptual basis for future creation of development environments for the machine learning paradigm. We used this conceptual tool in a case study of epilepsy seizure localization and where unintuitive solutions were discovered efficiently. Future development of such a conceptual design would allow for additional intelligent tools to aid the user in designing solutions and support reusability at the design level.

REFERENCES

1. Lindner, M., Vicente, R., Priesemann, V., & Wibral, M. (2011). TRENTOOL: A Matlab open source toolbox to analyse information flow in time series data with transfer entropy. *BMC neuroscience*, *12*(1), 119.

2. Vicente, R., Wibral, M., Lindner, M., & Pipa, G. (2011). Transfer entropy—a model-free measure of effective connectivity for the neurosciences. *Journal of computational neuroscience*, *30*(1), 45-67.

3. Wollstadt, P., Martinez-Zarzuela, M., Vicente, R., Diaz-Pernas, F. J., & Wibral, M. (2014). Efficient transfer entropy analysis of non-stationary neural time series. *PloS one*, *9*(7).

4. Rivolta, D., Heidegger, T., Scheller, B., Sauer, A., Schaum, M., Birkner, K., & Uhlhaas, P. J. (2015). Ketamine dysregulates the amplitude and connectivity of high-frequency oscillations in cortical–subcortical networks in humans: evidence from resting-state magnetoencephalography-recordings. *Schizophrenia bulletin*, *41*(5), 1105-1114.

5. Huang, C. S., Pal, N. R., Chuang, C. H., & Lin, C. T. (2015). Identifying changes in EEG information transfer during drowsy driving by transfer entropy. *Frontiers in human neuroscience*, *9*, 570.

6. Soriano, M. C., Niso, G., Clements, J., Ortín, S., Carrasco, S., Gudín, M., & Pereda, E. (2017). Automated detection of epileptic biomarkers in resting-state interictal MEG data. *Frontiers in neuroinformatics*, *11*, 43.

7. Wu, T., Chen, D., Chen, Q., Zhang, R., Zhang, W., Li, Y., & Zhang, J. (2018). Automatic lateralization of temporal lobe epilepsy based on MEG network features using support vector machines. *Complexity*, *2018*.

8. Téllez-Zenteno, J. F., & Hernández-Ronquillo, L. (2012). A review of the epidemiology of temporal lobe epilepsy. *Epilepsy research and treatment*, *2012*.

9. England, M. J., Liverman, C. T., Schultz, A. M., & Strawbridge, L. M. (2012). Epilepsy across the spectrum: Promoting health and understanding: A summary of the Institute of Medicine report. *Epilepsy & Behavior*, *25*(2), 266-276.

10. Strawbridge, L. M., Schultz, A. M., Liverman, C. T., & England, M. J. (Eds.). (2012). *Epilepsy across the spectrum: promoting health and understanding*. National Academies Press.

11. Ver Hoef, L. W., Sawrie, S., Killen, J., & Knowlton, R. C. (2008). Left mesial temporal sclerosis and verbal memory: a magnetoencephalography study. *Journal of Clinical Neurophysiology*, *25*(1), 1-6.

12. Coolen, T., Dumitrescu, A. M., Bourguignon, M., Wens, V., Urbain, C., & De Tiège, X. (2018). Presurgical electromagnetic functional brain mapping in refractory focal epilepsy. *Zeitschrift für Epileptologie*, *31*(3), 203-212.

13. Tamilia, E., Madsen, J. R., Grant, P. E., Pearl, P. L., & Papadelis, C. (2017). Current and emerging potential of magnetoencephalography in the detection and localization of high-frequency oscillations in epilepsy. *Frontiers in neurology*, *8*, 14.

14. Tadel, F., Baillet, S., Mosher, J. C., Pantazis, D., & Leahy, R. M. (2011). Brainstorm: a user-friendly application for MEG/EEG analysis. *Computational intelligence and neuroscience*, *2011*.

15. Puce, A., & Hämäläinen, M. S. (2017). A review of issues related to data acquisition and analysis in EEG/MEG studies. *Brain sciences*, *7*(6), 58.

16. Hämäläinen, M., Hari, R., Ilmoniemi, R. J., Knuutila, J., & Lounasmaa, O. V. (1993). Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of modern Physics*, *65*(2), 413.

17. Haneef, Z., Lenartowicz, A., Yeh, H. J., Levin, H. S., Engel Jr, J., & Stern, J. M. (2014). Functional connectivity of hippocampal networks in temporal lobe epilepsy. *Epilepsia*, *55*(1), 137-145.

18. Tan, A. C., & Gilbert, D. (2003). Ensemble machine learning on gene expression data for cancer classification.

19. Martelli, P. L., Fariselli, P., & Casadio, R. (2003). An ENSEMBLE machine learning approach for the prediction of all-alpha membrane proteins. *Bioinformatics*, *19*(suppl_1), i205-i211.

20. Jonsson, L., Borg, M., Broman, D., Sandahl, K., Eldh, S., & Runeson, P. (2016). Automated bug assignment: Ensemble-based machine learning in large scale industrial contexts. *Empirical Software Engineering*, *21*(4), 1533-1578.

21. Wang, S. J., Mathew, A., Chen, Y., Xi, L. F., Ma, L., & Lee, J. (2009). Empirical analysis of support vector machine ensemble classifiers. *Expert Systems with applications*, *36*(3), 6466-6476.

22. Aoe, J., Fukuma, R., Yanagisawa, T., Harada, T., Tanaka, M., Kobayashi, M., ... & Kishima, H. (2019). Automatic diagnosis of neurological diseases using MEG signals with a deep neural network. *Scientific reports*, *9*(1), 1-9.

23. Knowlton, R. C. (2006). The role of FDG-PET, ictal SPECT, and MEG in the epilepsy surgery evaluation. *Epilepsy & Behavior*, *8*(1), 91-101.

24. Pataraia, E., Simos, P. G., Castillo, E. M., Billingsley, R. L., Sarkari, S., Wheless, J. W., ... & Breier, J. I. (2004). Does magnetoencephalography add to scalp video-EEG as a diagnostic tool in epilepsy surgery?. *Neurology*, *62*(6), 943-948.

25. Youngerman, B. E., Khan, F. A., & McKhann, G. M. (2019). Stereoelectroencephalography in epilepsy, cognitive neurophysiology, and psychiatric disease: safety, efficacy, and place in therapy. *Neuropsychiatric disease and treatment*, *15*, 1701.

26. Quesney, L. F. (1992). Extratemporal epilepsy: Clinical presentation, pre-operative EEG localization and surgical outcome. *Acta Neurologica Scandinavica*, *86*(S140), 81-94.

27. Das, M., Cui, R., Campbell, D. R., Agrawal, G., & Ramnath, R. (2015, October). Towards methods for systematic research on big data. In *2015 IEEE International Conference on Big Data (Big Data)* (pp. 2072-2081). IEEE.

28. Callebaut, W. (2012). Scientific perspectivism: A philosopher of science's response to the challenge of big data biology. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*, *43*(1), 69-80.

29. Kwiatkowska, M., & Atkins, A. S. (2005, December). Integrating knowledge-driven and data-driven approaches for the derivation of clinical prediction rules. In *Fourth International Conference on Machine Learning and Applications (ICMLA'05)* (pp. 6-pp). IEEE.

30. Mestre, T. A., Eberly, S., Tanner, C., Grimes, D., Lang, A. E., Oakes, D., & Marras, C. (2018). Reproducibility of data-driven Parkinson's disease subtypes for clinical research. *Parkinsonism & related disorders*, *56*, 102-106.

31. Sanchez-Pinto, L. N., Luo, Y., & Churpek, M. M. (2018). Big data and data science in critical care. *Chest*, *154*(5), 1239-1248.

32. Hemingway, H., Asselbergs, F. W., Danesh, J., Dobson, R., Maniadakis, N., Maggioni, A. & Anker, S. D. (2018). Big data from electronic health records for

early and late translational cardiovascular research: challenges and potential. *European heart journal*, *39*(16), 1481-1495.

33. Brunton, B. W., & Beyeler, M. (2019). Data-driven models in human neuroscience and neuroengineering. *Current opinion in neurobiology*, *58*, 21-29.

34. Hulsen, T., Jamuar, S. S., Moody, A., Karnes, J. H., Orsolya, V., Hedensted, S., ... & McKinney, E. (2019). From big data to precision medicine. *Frontiers in medicine*, *6*, 34.

35. Montáns, F. J., Chinesta, F., Gómez-Bombarelli, R., & Kutz, J. N. (2019). Data-driven modeling and learning in science and engineering. *Comptes Rendus Mécanique*, *347*(11), 845-855.

36. Johnson, J. W. (2020, October). Benefits and Pitfalls of Jupyter Notebooks in the Classroom. In *Proceedings of the 21st Annual Conference on Information Technology Education* (pp. 32-37).

37. Pimentel, J. F., Murta, L., Braganholo, V., & Freire, J. (2021). Understanding and improving the quality and reproducibility of Jupyter notebooks. *Empirical Software Engineering*, *26*(4), 1-55.

38. DePratti, R. (2019). Using Jupyter notebooks in a big data programming course. *Journal of Computing Sciences in Colleges*, *34*(6), 157-159.

39. Perkel, J. M. (2018). Why Jupyter is data scientists' computational notebook of choice. *Nature*, *563*(7732), 145-147.

40. Fradkov, A. L. (2020). Early history of machine learning. *IFAC-PapersOnLine*, *53*(2), 1385-1390.

41. Rosenblatt, F. (1960). Perceptron simulation experiments. *Proceedings of the IRE*, *48*(3), 301-309.

42. Rosenow, F., & Lüders, H. (2001). Presurgical evaluation of epilepsy. *Brain*, *124*(9), 1683-1700.

APPENDIX A: IRB APPROVAL

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM
Office of the Institutional Review Board for Human Use

470 Administration Building
701 20th Street South
Birmingham, AL 35294-0104
205.934.3789 | Fax 205.934.1301 | irb@uab.edu

APPROVAL LETTER

TO: Mohamed, Ismail S

FROM: University of Alabama at Birmingham Institutional Review Board
Federalwide Assurance # FWA00005960
IORG Registration # IRB00000196 (IRB 01)
IORG Registration # IRB00000726 (IRB 02)

IORG Registration # IRB00012550 (IRB 03)

DATE: 13-Oct-2020

RE: IRB-300000476
Non-invasive Delineation of the Spatial Extent of the Epileptogenic Zone in Focal Epilepsy

---

The IRB reviewed and approved the Continuing Review submitted on 12-Oct-2020 for the above referenced project. The review was conducted in accordance with UAB's Assurance of Compliance approved by the Department of Health and Human Services.

Type of Review:	Expedited
Expedited Categories:	5,
Determination:	Approved
Approval Date:	13-Oct-2020
Approval Period:	One Year
Expiration Date:	12-Oct-2021

The following populations are approved for inclusion in this project:

- Children – CRL 1

The following apply to this project related to informed consent and/or assent:

- Waiver of HIPAA
- Waiver of Informed Consent

Documents Included in Review:

- ipr.201010.pdf

Record Number
IRB-300000476
Done Save

Non-invasive Delineation of the Spatial Extent of the Epileptogenic Zone in Focal Epilepsy
Ismail S Mohamed - Ped - Neurology

Human Subjects
View Mode - See Details
Change Project Info

Submissions (10) Linkages Summaries ?

Home Submissions Personnel Amendment Personnel

Submission
Reviews (1)
Personnel (4)

Research Personnel

Add Update Summary

All Certifications and Training

PI	Name	COI	Start Date		End Date
	Ismail Mohamed - Ped - Neurology Role: PI	✔	28-Aug-2017	Retire	Remove
	Certifications and Training	Responsible Person		CV Email	
	Anthony Bowman - Graduate School Dean's Office Role: Subinvestigator	✔	14-Apr-2021	Retire	Remove
	Certifications and Training	Responsible Person		CV Email	
	Pravinkumar Kandhare - Electrical & Computer Engineering Role: Subinvestigator	✔	08-Feb-2021	Retire	Remove
	Certifications and Training	Responsible Person		CV Email	
	Jeffery Killen - Admin-Operations-TKC Role: Other Personnel	✔	27-Nov-2017	Retire	Remove
	Certifications and Training	Responsible Person		CV Email	

APPENDIX B: TABULATED SUPPLEMENTARY RESULTS

Table 1: EEG only, standard 10 fold cross validation

Algorithm	f-measure	rightMesTemp	leftMesTemp	Actual
BayesNet	0.818	5	1	rightMesTemp
		1	4	leftMesTemp
NaiveBayes	1.000	6	0	rightMesTemp
		0	5	leftMesTemp
NaiveBayesUpdateable	1.000	6	0	rightMesTemp
		0	5	leftMesTemp
Logistic Regression	1.000	6	0	rightMesTemp
		0	5	leftMesTemp
SGD	1.000	6	0	rightMesTemp
		0	5	leftMesTemp
Multilayer Perceptron	1.000	6	0	rightMesTemp
		0	5	leftMesTemp
SimpleLogistic	0.818	5	1	rightMesTemp
		1	4	leftMesTemp
SMO (SVM)	0.909	5	1	rightMesTemp
		0	5	leftMesTemp
DecisionStump	0.818	5	1	rightMesTemp
		1	4	leftMesTemp
J48	0.723	5	1	rightMesTemp
		2	3	leftMesTemp
LMT (log tree)	0.818	5	1	rightMesTemp
		1	4	leftMesTemp
Random Forest	0.818	5	1	rightMesTemp
		1	4	leftMesTemp
Random Tree	0.636	4	2	rightMesTemp
		2	3	leftMesTemp

Table 2: MEG only

Algorithm	f-measure	rightMesTemp	leftMesTemp	Actual
BayesNet	0.727	4	2	rightMesTemp
		1	4	leftMesTemp
NaiveBayes	0.273	2	4	rightMesTemp
		4	1	leftMesTemp
NaiveBayesUpdateable	0.273	2	4	rightMesTemp
		4	1	leftMesTemp
Logistic Regression	0.545	3	3	rightMesTemp
		2	3	leftMesTemp
SGD	0.545	3	3	rightMesTemp
		2	3	leftMesTemp

Multilayer Perceptron	0.545	3	3	rightMesTemp
		2	3	leftMesTemp
SimpleLogistic	0.723	5	1	rightMesTemp
		2	3	leftMesTemp
SMO (SVM)	0.545	3	3	rightMesTemp
		2	3	leftMesTemp
DecisionStump	0.545	3	3	rightMesTemp
		2	3	leftMesTemp
J48	?	6	0	rightMesTemp
		5	0	leftMesTemp
LMT (log tree)	0.723	5	1	rightMesTemp
		2	3	leftMesTemp
Random Forest	0.630	3	3	rightMesTemp
		1	4	leftMesTemp
Random Tree	0.545	3	3	rightMesTemp
		2	3	leftMesTemp

Table 3: EEG and MEG together

Algorithm	f-measure	rightMesTemp	leftMesTemp	Actual
BayesNet	0.727	4	2	rightMesTemp
		1	4	leftMesTemp
NaiveBayes	0.273	2	4	rightMesTemp
		4	1	leftMesTemp
NaiveBayesUpdateable	0.273	2	4	rightMesTemp
		4	1	leftMesTemp
Logistic Regression	0.723	5	1	rightMesTemp
		2	3	leftMesTemp
SGD	0.908	6	0	rightMesTemp
		1	4	leftMesTemp
Multilayer Perceptron	0.908	6	0	rightMesTemp
		1	4	leftMesTemp
SimpleLogistic	0.617	5	1	rightMesTemp
		3	2	leftMesTemp
SMO (SVM)	0.908	6	0	rightMesTemp
		1	4	leftMesTemp

DecisionStump	0.445	2	4	rightMesTemp
		2	3	leftMesTemp
J48	0.723	5	1	rightMesTemp
		2	3	leftMesTemp
LMT (log tree)	0.617	5	1	rightMesTemp
		3	2	leftMesTemp
Random Forest	0.630	3	3	rightMesTemp
		1	4	leftMesTemp
Random Tree	0.636	4	2	rightMesTemp
		2	3	leftMesTemp

Table 1: Right vs Left mesial temporal with EEG only, standard 10 fold cross validation

Algorithm	f-measure	rightMesTemp	leftMesTemp	Actual
BayesNet	0.636	4	2	rightMesTemp
		2	3	leftMesTemp
NaiveBayes	0.909	5	1	rightMesTemp
		0	5	leftMesTemp
NaiveBayesMultinomial	0.091	1	5	rightMesTemp
		5	0	leftMesTemp
Logistic Regression	0.815	4	2	rightMesTemp
		0	5	leftMesTemp
SGD	0.909	5	1	rightMesTemp
		0	5	leftMesTemp
Multilayer Perceptron	**1.000**	**6**	**0**	**rightMesTemp**
		0	**5**	**leftMesTemp**
SimpleLogistic	0.815	4	2	rightMesTemp
		0	5	leftMesTemp
SMO (SVM)	0.909	5	1	rightMesTemp
		0	5	leftMesTemp
DecisionStump	0.818	5	1	rightMesTemp
		1	4	leftMesTemp

J48	0.538	4	2	rightMesTemp
		3	2	leftMesTemp
LMT (log tree)	0.727	4	2	rightMesTemp
		1	4	leftMesTemp
Random Forest	0.723	5	1	rightMesTemp
		2	3	leftMesTemp
Random Tree	0.909	5	1	rightMesTemp
		0	5	leftMesTemp

Table 2: Right vs Left mesial temporal with MEG only, standard 10 fold cross validation

Algorithm	f-measure	rightMesTemp	leftMesTemp	Actual
BayesNet	0.630	3	3	rightMesTemp
		1	4	leftMesTemp
NaiveBayes	0.630	3	3	rightMesTemp
		1	4	leftMesTemp
NaiveBayesMultinomial	?	6	0	rightMesTemp
		5	0	leftMesTemp
Logistic Regression	**0.818**	**5**	**1**	rightMesTemp
		1	**4**	leftMesTemp
SGD	0.545	3	3	rightMesTemp
		2	3	leftMesTemp
Multilayer Perceptron	0.727	4	2	rightMesTemp
		1	4	leftMesTemp
SimpleLogistic	**0.818**	**5**	**1**	rightMesTemp
		1	**4**	leftMesTemp
SMO (SVM)	0.727	4	2	rightMesTemp
		1	4	leftMesTemp
DecisionStump	0.545	3	3	rightMesTemp
		2	3	leftMesTemp
J48	?	6	0	rightMesTemp
		5	0	leftMesTemp

LMT (log tree)	**0.818**	**5**	**1**	rightMesTemp
		1	**4**	leftMesTemp
Random Forest	0.727	4	2	rightMesTemp
		1	4	leftMesTemp
Random Tree	0.636	4	2	rightMesTemp
		2	3	leftMesTemp

Table 3: Right vs Left mesial temporal with EEG and MEG, standard 10 fold cross validation

Algorithm	f-measure	rightMesTemp	leftMesTemp	Actual
BayesNet	0.630	3	3	rightMesTemp
		1	4	leftMesTemp
NaiveBayes	0.630	3	3	rightMesTemp
		1	4	leftMesTemp
NaiveBayesMultinomial	0.091	1	5	rightMesTemp
		5	0	leftMesTemp
Logistic Regression	0.818	5	1	rightMesTemp
		1	4	leftMesTemp
SGD	**0.908**	**6**	**0**	**rightMesTemp**
		1	**4**	**leftMesTemp**
Multilayer Perceptron	0.818	5	1	rightMesTemp
		1	4	leftMesTemp
SimpleLogistic	0.494	5	1	rightMesTemp
		4	1	leftMesTemp
SMO (SVM)	0.818	5	1	rightMesTemp
		1	4	leftMesTemp
DecisionStump	0.727	4	2	rightMesTemp
		1	4	leftMesTemp
J48	0.538	4	2	rightMesTemp
		3	2	leftMesTemp

LMT (log tree)	0.494	5	1	rightMesTemp
		4	1	leftMesTemp
Random Forest	0.727	4	2	rightMesTemp
		1	4	leftMesTemp
Random Tree	0.630	3	3	rightMesTemp
		1	4	leftMesTemp

Table 4: Right temporal vs. bilateral extra-temporal with EEG only, standard 10 fold cross validation

Algorithm	f-measure	rightMesTemp	leftMesTemp	Actual
BayesNet	0.491	0	6	rightMesTemp
		4	11	Extra-temporal
NaiveBayes	0.681	4	2	rightMesTemp
		5	10	Extra-temporal
NaiveBayesMultinomial	?	0	6	rightMesTemp
		0	15	Extra-temporal
Logistic Regression	0.524	1	5	rightMesTemp
		5	10	Extra-temporal
SGD	0.619	2	4	rightMesTemp
		4	11	Extra-temporal
Multilayer Perceptron	0.626	1	5	rightMesTemp
		2	13	Extra-temporal
SimpleLogistic	0.546	0	6	rightMesTemp
		2	13	Extra-temporal
SMO (SVM)	0695	2	4	rightMesTemp
		2	13	Extra-temporal
DecisionStump	0.524	1	5	rightMesTemp
		5	10	Extra-temporal
J48	0.674	3	3	rightMesTemp

		4	11	Extra-temporal
LMT (log tree)	0.546	0	6	rightMesTemp
		2	13	Extra-temporal
Random Forest	0.593	1	5	rightMesTemp
		3	12	Extra-temporal
Random Tree	0.449	1	5	rightMesTemp
		7	8	Extra-temporal

Table 5: Right temporal vs. bilateral extra-temporal with EEG and MEG, standard 10 fold cross validation

Algorithm	f-measure	rightMesTemp	leftMesTemp	Actual
BayesNet	0.657	3	3	rightMesTemp
		4	10	Extra-temporal
NaiveBayes	0.300	3	3	rightMesTemp
		11	3	Extra-temporal
NaiveBayesMultinomial	?	0	6	rightMesTemp
		0	14	Extra-temporal
Logistic Regression	0.657	3	3	rightMesTemp
		4	10	Extra-temporal
SGD	0.657	3	3	rightMesTemp
		4	10	Extra-temporal
Multilayer Perceptron	0.600	2	4	rightMesTemp
		4	10	Extra-temporal
SimpleLogistic	0.600	2	4	rightMesTemp
		4	10	Extra-temporal
SMO (SVM)	0613	3	3	rightMesTemp
		5	9	Extra-temporal

DecisionStump	0.567	4	2	rightMesTemp
		7	7	Extra-temporal
J48	0.461	1	5	rightMesTemp
		6	8	Extra-temporal
LMT (log tree)	0.600	2	4	rightMesTemp
		4	10	Extra-temporal
Random Forest	0.609	1	5	rightMesTemp
		2	12	Extra-temporal
Random Tree	0.425	2	4	rightMesTemp
		8	6	Extra-temporal

Table 6: Left temporal vs. bilateral extra-temporal with EEG only, standard 10 fold cross validation

Algorithm	f-measure	rightMesTemp	leftMesTemp	Actual
BayesNet	0.583	0	5	rightMesTemp
		3	13	Extra-temporal
NaiveBayes	0.861	4	1	rightMesTemp
		2	14	Extra-temporal
NaiveBayesMultinomial	?	0	5	rightMesTemp
		0	16	Extra-temporal
Logistic Regression	**0.910**	**5**	**0**	**rightMesTemp**
		2	**14**	**Extra-temporal**
SGD	**0.954**	**5**	**0**	**rightMesTemp**
		1	**15**	**Extra-temporal**
Multilayer Perceptron	**0.954**	**5**	**0**	**rightMesTemp**
		1	**15**	**Extra-temporal**

SimpleLogistic	0.769	3	2	rightMesTemp
		3	13	Extra-temporal
SMO (SVM)	0.810	3	2	rightMesTemp
		2	14	Extra-temporal
DecisionStump	0.676	2	3	rightMesTemp
		4	12	Extra-temporal
J48	0.457	0	5	rightMesTemp
		7	9	Extra-temporal
LMT (log tree)	0.769	3	2	rightMesTemp
		3	13	Extra-temporal
Random Forest	0.791	2	3	rightMesTemp
		1	15	Extra-temporal
Random Tree	0.729	3	2	rightMesTemp
		4	12	Extra-temporal

Table 7: Left temporal vs. bilateral extra-temporal with EEG and MEG, standard 10 fold cross validation

Algorithm	f-measure	leftMesTemp	Extra-temporal	Actual
BayesNet	failure	?	?	leftMesTemp
		?	?	Extra-temporal
NaiveBayes	0.535	5	0	leftMesTemp
		10	6	Extra-temporal
NaiveBayesMultinomial	?	0	5	leftMesTemp
		0	14	Extra-temporal
Logistic Regression	0.729	3	2	leftMesTemp
		4	12	Extra-temporal
SGD	**0.905**	**4**	**1**	**leftMesTemp**
		1	**15**	**Extra-**

				temporal
Multilayer Perceptron	0.810	3	2	leftMesTemp
		2	14	Extra-temporal
SimpleLogistic	0.896	3	2	leftMesTemp
		0	16	Extra-temporal
SMO (SVM)	0.851	3	2	leftMesTemp
		1	15	Extra-temporal
DecisionStump	0.861	4	1	leftMesTemp
		2	14	Extra-temporal
J48	0.457	0	5	leftMesTemp
		7	9	Extra-temporal
LMT (log tree)	0.896	3	2	leftMesTemp
		0	16	Extra-temporal
Random Forest	0.833	2	3	leftMesTemp
		0	16	Extra-temporal
Random Tree	0.810	3	2	leftMesTemp
		2	14	Extra-temporal

Table 8: Right vs Left mTLE, standard 10 fold cross validation using the second data file

Algorithm	Beta & low gamma EEG (1st run)	Theta EEG	Theta, Beta, Gamma EEG	EEG&MEG
BayesNet	0.818	0.494	0.494	**0.908**
NaiveBayes	**1.000**	0.636	0.617	0.261
NaiveBayesUpdateable	**1.000**	0.636	0.617	0.291
Multinomial				
Logistic Regression	**1.000**	0.617	0.617	0.818
SGD	**1.000**	0.617	0.538	0.545

Multilayer Perceptron	**1.000**	0.617	0.538	0.545
SimpleLogistic	**0.818**	0.425	0.636	0.727
SMO (SVM)	**0.909**	0.353	0.273	0.545
DecisionStump	**0.818**	0.723	0.723	0.727
J48	**0.723**	0.545	0.545	0.545
LMT (log tree)	**0.818**	0.425	0.636	0.727
Random Forest	0.818	0.455	0.636	**0.908**
Random Tree	0.636	0.273	0.617	**0.727**

Table 9: Right vs Left Frontal, standard 5 fold cross validation

Algorithm	Delta EEG	Theta EEG	Beta and low gamma EEG	Delta, theta, beta, low gamma EEG	Delta, theta, beta, low gamma MEG	EEG&MEG
BayesNet	?	?	0.556	0.476	0.778	0.778
NaiveBayes	0.556	0.410	0.738	?	0.778	0.778
NaiveBayesMultinomial	?	?	0.667	?	?	?
Logistic Regression	0.646	0.556	**1.000**	0.778	**0.882**	**0.882**
SGD	0.646	0.476	**1.000**	0.738	0.738	0.738
Multilayer Perceptron	0.556	0.556	**1.000**	0.882	**0.882**	**0.882**
SimpleLogistic	0.459	0.567	0.556	0.646	0.646	0.646
SMO (SVM)	0.410	?	**0.892**	**0.882**	**0.882**	**0.882**
DecisionStump	0.444	0.459	0.444	0.459	0.556	0.556

J48	0.459	0.459	0.783	0.778	?	?
LMT (log tree)	0.459	0.567	0.556	0.646	0.646	0.646
Random Forest	0.410	0.476	0.459	0.556	0.646	0.646
Random Tree	0.410	0.410	0.556	0.778	0.410	0.410

Table 10: Right vs Left Frontal, standard 5 fold cross validation

Algorithm	Theta MEG	Beta MEG	low gamma MEG	Beta, low gamma MEG	theta, beta, low gamma MEG
BayesNet	0.646	0.778	0.778	0.778	0.778
NaiveBayes	0.778	**0.882**	0.738	**0.882**	0.778
NaiveBayesMultinomial	?	?	?	?	?
Logistic Regression	?	**0.882**	**0.882**	**0.882**	0.778
SGD	?	0.778	**0.882**	**0.882**	0.738
Multilayer Perceptron	?	0.738	**0.882**	**0.882**	0.738
SimpleLogistic	0.778	0.738	**0.882**	**0.882**	0.778
SMO (SVM)	?	**0.882**	**0.882**	**0.882**	0.738
DecisionStump	0.675	0.778	**0.882**	0.778	0.675
J48	?	?	?	?	?
LMT (log tree)	0.778	0.738	**0.882**	**0.882**	0.778
Random Forest	0.738	0.476	0.738	0.778	0.778
Random Tree	0.646	0.646	0.778	0.778	0.567

Table 11: Left frontal vs bilateral extra-frontal, standard 10 fold cross validation

Algorithm	Delta EEG	Theta EEG	Beta and low gamma EEG	Delta, theta, beta, low gamma EEG	Delta, theta, beta, low gamma MEG	EEG&MEG
BayesNet	?	0.777	?	0.777	0.872	0.872
NaiveBayes	0.653	0.753	0.777	0.777	0.291	0.347
NaiveBayesMultinomial	?	?	?	?	?	?
Logistic Regression	0.727	0.727	0.818	0.753	**0.951**	0.836
SGD	0.777	0.777	**0.909**	0.777	**0.951**	**0.951**
Multilayer Perceptron	0.777	0.777	**0.951**	0.777	**0.909**	**0.909**
SimpleLogistic	0.753	0.777	**0.889**	0.777	**0.951**	**0.951**
SMO (SVM)	?	0.777	**?**	**?**	**0.951**	**0.909**
DecisionStump	0.753	0.753	0.777	0.753	0.872	0.872
J48	?	0.753	**0.889**	0.786	?	0.786
LMT (log tree)	0.753	0.777	0.889	0.777	**0.951**	**0.951**
Random Forest	?	0.777	?	?	0.777	0.852
Random Tree	0.700	0.727	**0.909**	0.727	0.852	0.818

Table 12: Right frontal vs bilateral extra-frontal, standard 10 fold cross validation

Algorithm	Delta EEG	Theta EEG	Beta and low gamma EEG	Delta, theta, beta, low gamma EEG	Delta, theta, beta, low gamma MEG	EEG&MEG
BayesNet	?	?	?	?	**0.805**	0.805

NaiveBayes	0.084	0.364	0.611	0.261	0.287	0.287
NaiveBayesMultinomial	?	?	?	?	?	?
Logistic Regression	0.570	0.613	0.398	0.600	**0.818**	0.636
SGD	0.590	0.590	0.745	0.636	0.778	0.600
Multilayer Perceptron	0.513	0.484	0.422	0.600	0.727	0.579
SimpleLogistic	?	0.566	?	?	0.566	?
SMO (SVM)	?	0.590	**?**	0.642	0.611	0.566
DecisionStump	0.513	0.455	0.590	0.590	**0.805**	0.805
J48	0.540	0.540	0.513	0.485	?	0.485
LMT (log tree)	?	0.513	?	?	0.566	?
Random Forest	0.540	0.579	0.566	0.540	0.642	0.745
Random Tree	0.438	0.600	0.485	0.441	0.611	0.745

Table 12: Left mesial temporal vs right mesial temporal, 10 fold cross validation

Algorithm	Theta MEG	Theta EEG	Beta, Gamma EEG	Theta, Beta, Gamma EEG	EEG&MEG
BayesNet	0.646	0.494	0.818	0.636	**0.908**
NaiveBayes	0.778	0.636	**1.000**	**0.909**	0.261
NaiveBayesMultinomial	?	0.636	0.091	0.091	0.291
Logistic Regression	?	0.617	**1.000**	0.815	0.818

SGD	?	0.617	**1.000**	**0.909**	0.545
Multilayer Perceptron	?	0.617	**1.000**	**1.000**	0.545
SimpleLogistic	0.778	0.425	0.818	0.815	0.727
SMO (SVM)	?	0.353	**0.909**	**0.909**	0.545
DecisionStump	0.675	0.723	0.818	0.818	0.727
J48	?	0.545	0.723	0.538	0.545
LMT (log tree)	0.778	0.425	0.818	0.727	0.727
Random Forest	0.738	0.455	0.818	0.723	**0.908**
Random Tree	0.675	0.273	0.636	**0.909**	0.727

Table 13: Left vs right TLE, 10 fold cross validation, MEG only

Algorithm	Theta Coherence	Alpha Coherence	Beta Coherence	Low Gamma Coherence	Beta & Low Gamma Coherence	TABG Coherence
BayesNet	0.364	0.364	0.723	0.723	0.696	0.617
NaiveBayes	0.696	0.617	0.538	0.545	0.364	0.538
NaiveBayesMultinomial	0.091	0.636	0.364	0.455	0.538	0.445

Logistic Regression	0.445	0.331	0.723	0.445	0.636	0.455
SGD	0.331	0.261	0.538	0.636	0.723	0.723
Multilayer Perceptron	0.331	0.261	0.617	0.617	0.636	OOM
SimpleLogistic	0.140	0.195	**0.818**	0.445	0.727	0.331
SMO (SVM)	0.261	0.261	0.617	0.617	0.636	0.455
DecisionStump	0.331	0.140	0.727	0.445	0.727	0.331
J48	0.636	0.261	0.727	0.445	0.727	0.445
LMT (log tree)	0.140	0.195	**0.818**	0.445	0.727	0.331
Random Forest	0.331	0.364	0.808	0.455	0.696	0.636
Random Tree	0.445	0.455	0.723	0.455	**0.818**	0.538