Federated Machine Learning for Translational Research

Completed Research

Manoj A. Thomas
Virginia Commonwealth University
mthomas@vcu.edu

Diya S. Abraham
Techies Without Borders
diyasuzanne@gmail.com

Dapeng Liu
Virginia Commonwealth University
Liud22@vcu.edu

Abstract

Translational research (TR) is the harnessing of knowledge from basic science and clinical research to advance healthcare. As a sister discipline, Translational informatics (TI) concerns the application of informatics theories, methods, and frameworks to TR. This research builds upon TR concepts, and aims to bring advances in machine learning (ML) and data analytics for improving clinical decision support. A federated machine learning (FML) architecture is proposed to aggregate multiple sources, and intermediate data analytic processes and products to output high quality knowledge discovery and decision making. The proposed architecture is evaluated for its operational performance based on three propositions, and a case for clinical decision support in the prediction of adult Sepsis is presented. Our research illustrates how IS scholarship may provide valuable contributions to the advancement of TI.

Keywords

Federated machine learning, translational research, translational informatics, clinical decision support

Introduction

Machine Learning (ML) consists of algorithms for data-driven predictions or decisions. Well known applications of ML include speech recognitions, recommender systems, and more recently, autonomous vehicles (Kuderer, Gulati, & Burgard, 2015). Traditionally, ML involves the use of powerful computational platforms for applying data mining techniques and learning algorithms on large datasets.

In healthcare, many realistic scenarios involve smaller datasets and rare events, and therefore traditional ML is not as effective in such contexts. A preferred approach is to have algorithms that can interact with agents and optimize learning through these interactions, where the agents can themselves be human (Holzinger, 2016). This implies that in scenarios where datasets are relative small, establishing a symbiotic relationship between the computational and human agents is more relevant than the traditional approach of relying on large datasets.

Human agents using edge devices such as mobile phones and wearable technologies are examples of user interactions that can be used in ML. Furthermore, user interactions have the advantage of natural labelling of data thus reducing a search space of exponential possibilities drastically by heuristic selection of samples (Holzinger, 2016).

Traditional systems that support ML are not optimized for leveraging human agents. The common approach is to consolidate datasets on a server or cloud on which training in conducted and then the algorithms are productionalized. However, this approach has setbacks. It assumes ubiquitous access to the cloud, continuous streaming of data, high throughput, and network dependence. Additionally, the models stored in the cloud tend to be highly generalized as it does not take into consideration the imbalance in user interactions. In this research, we propose a federated ML (FML) approach to address these challenges. Federated machine learning (FML) is the pipelining of machine learning and data analytic tools by incorporating multiple source datasets, and intermediate data analytic processes and products to output high quality results that support knowledge discovery and decision making. It aims to move ML closer to the end user. This paper will demonstrate how FML may be applied in the context of translational research (TR).

TR is the harnessing of knowledge from basic science and clinical research to develop new drugs, devices, and treatment options (Sussman, Valente, Rohrbach, Skara, & Ann Pentz, 2006; Woolf, 2008). National Institute of Health identifies two related cycles of TR (Payne, Embi, & Sen, 2009; Sung et al., 2003). First, it is the translation of discoveries from the laboratory to clinical research involving human trials and studies. Second, it is the translation of findings from clinical research to enhance best practices. The cycles thus involve the translation of knowledge and evidence from the “bench” (i.e., the laboratory) to the “bedside” (i.e., clinical and public health practice), and reciprocally from the “bedside” to the “bench” (Payne et al., 2009; Woolf, 2008). Translational informatics (TI) is a sister domain to TR, and concerned with the application of informatics theories and methods to TR.

Rapid advances in machine learning (ML) and data analytics provide novel opportunities to apply FML in TI. Doing so may improve the quality of interaction of models with human agency, identify and gather data from rare and unique events, and limit dependency on cloud based services. FML could bring computational and analytical solutions closer to the decision maker.

This study aims to achieve three objectives. First, we build upon TR concepts to propose an architecture for FML. We do so by highlighting why current cloud-client approaches are not suited for FML. Second, we evaluate the proposed architecture for its operational performance based on three propositions. Third, we envisage how FML may facilitate knowledge discovery and clinical decision support in TI. This study makes two contributions. First, our research demonstrates that IS scholarship has much to contribute to the advancement of disciplines such as TR and bioinformatics. Based on the knowledge discovery and clinical decision support needs of TR (Payne et al., 2009; Sung et al., 2003), we used design science methodology to develop a platform to federate ML. The proposed platform leverages low power servers to mediate the execution of the ML algorithms. Second, the study represents a natural progression in the adaptation and use of IS methods, models, and frameworks to address many data and knowledge challenges in TI. By presenting a case for clinical decision support in the prediction of adult Sepsis, we show how socio-technical systems that incorporate machine learning and data analytic tools are applicable for knowledge discovery and decision making.

The rest of this study is organized as follows. We first provide a background on TR and TI, including a review of related literature. We then highlight the limitations of the current cloud-client approach for ML, and present an architecture for FML. Based on three propositions, we then conduct an evaluation of the operational performance of the proposed architecture. The practical and theoretical contributions are discussed and we conclude by paper by presenting directions for future research.

Background and Concepts

Translational Research (TR)

TR is crucial to the healthcare industry. Literature provides many definitions of TR across various disciplines (Westfall, Mold, & Fagnan, 2007). TR is commonly represented in two stages. The first stage is translating knowledge, techniques or mechanisms from basic sciences to the development of new treatments. The second stage is translating findings from clinical trials to clinical and public health practice (Woolf, 2008). Often, a reciprocal stage is also included in TR, where observations from practice is intended to inform basic science (Figure 1). In a nutshell, TR is the interface between basic science and clinical practice, the goal of which is to produce a promising treatment that can be “brought to market” and ultimately to improve public health (Woolf, 2008). It is the effective translation of advances in basic science into new approaches for the prevention, diagnosis, and treatment of diseases. TR is of critical importance and essential to improving health (Fontanarosa & DeAngelis, 2002). TR is an established priority at many leading scientific institutions such the National Institutes of Health (NIH), research universities, and health systems engaged in biomedical research programs.

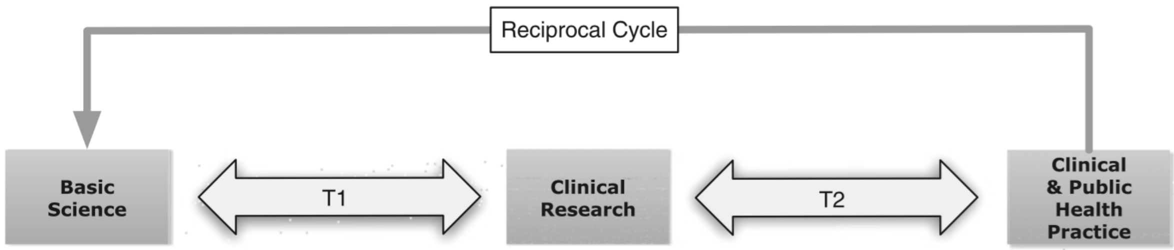

Figure 1. Translational Research Cycle (Payne et al., 2009; Sung et al., 2003)

Translational Informatics (TI)

TI focuses on the application of informatics theories, emergent technologies, and socio-technical methods for TR (Chen, Qian, Yan, & Shen, 2013). TI enables the 1) acquisition of knowledge and information, 2) representation of knowledge and information in an actionable format, 3) semantic integration and aggregation of data to support the discovery and validation of knowledge models, 4) dissemination of knowledge through public engagement (Harris et al., 2009; Payne et al., 2009). The importance of TI in drug development, preclinical trials, and clinical decision support is well recognized (Rubio et al., 2010). Pharmaceutical industries heavily invest in TI to advance new tools and devices applicable to healthcare (Beaulah, Correll, Munro, & Sheldon, 2008; Lee et al., 2000). Nevertheless, research is sparse on effective approaches and strategies to support its multiple functions.

Surprisingly, not much research has been published in IS related to TI. To better understand the role of IS in addressing challenges related to TI, we conducted a detailed review of literature. We started with a full-text search of the PROQUEST database to identify studies published in journals from the IS journal ranking page of the Association of Information Systems (AIS). We searched for the terms "Translational Research" or "Translational Informatics." Within IS, we were interested in studies that have adopted the design science methodology. Therefore, we further filtered the results using the search term "Design Science." Our search yielded 27 studies published between 2002 and 2017, among which 19 were dissertations and thesis. Among the remaining eight articles, only one study (Gholami, Watson, Molla, Hasan, & Bjørn-Andersen, 2016) was published in an IS journal. This research, published in the Journal of the Association of Information Systems, merely mentions that IS research could follow the tradition of TR to test and publish discoveries beneficial to practice. It does not develop or evaluate DS artifacts related to TI. In the TI domain, we found many studies that have adopted and used IS methods, models, and frameworks. Examples include pattern recognition (Baldi & Long, 2001; Pan, 2002), data mining (Becker et al., 2003), data visualization (Porollo, Adamczak, & Meller, 2004), data representation and annotation (Koski, Gray, Lang, & Burger, 2005), and decision-making (Beck et al., 2008; Castaneda et al., 2015; Lindgren, 2008; O'neill, Dluhy, & Chin, 2005). Our review of literature indicates that there is much potential for applying IS scholarship in the advancement of TI. In this research, we present one such case where machine learning, data analytics, and cloud computing are for clinical decision support.

The Need for FML

In the healthcare industry, the prevalent system architecture is the client/server model (Coulouris, Dollimore, & Kindberg, 2005; Fielding & Taylor, 2000). With the growing popularity of the cloud-based solutions, the emergent model of system architecture aligns more towards variations of cloud-related services such as SaaS, PaaS, and DaaS (Newman, 2015; Singleton, 2016). In this new paradigm of cloud-based computing, ML models are stored and managed in the cloud (Li, Thomas, & Osei-Bryson, 2017) which are then accessed and used by the application on the end user's device. By contrast, the objective of our proposed architecture is to move machine learning closer to edge devices and end users, and thereby relying less on constant access to massive data and intelligence stored in the cloud.

Payne et al. (2009) proposed a conceptual framework to incorporate major categories of information collected, managed and disseminated through TR. Specifically, it targets the integration of information and data spanning distinct users, data sources, and evidence silos. Modeling and machine learning approaches may then be used for the analysis of data and extraction of knowledge for clinical decision support. The framework emphasizes the use of cutting-edge technology and socio-technical approaches in TI research. We use this framework as the foundation to propose an architecture for FML.

Our review of literature did not find any research that has addressed FML. The closest reference is a Google Research Blog on collaborative machine learning without centralized training data (McMahan & Ramage, 2017). The blog entry refers to the use of Tensorflow technology to enable mobile phones collaboratively learn a shared prediction model. The approach suggests decoupling ML from data stored in the cloud. Although details presented in the blog entry are vague, it is ascertained that the proposed federated learning approach is optimized for modern high-end Android devices, and enables smarter ML models, lower latency, and less power consumption.

As indicated earlier, TI is at a nascent stage when it comes to applying newer technologies such as ML and data analytics. Current industry practice is the use of cloud-client architecture model, which is heavily network dependent and requires integrated data models and real-time communication with the cloud. In Table 1, we evaluate this practice on dimensions such as architecture, service model, access, machine learning, data model, and security. The table also presents preferred approach for FML across these dimensions.

Dimension	Industry Practice (Current Approach)	Shortcomings of Current Approach	Preferred Approach
Architecture	Cloud/client model	Constant cloud dependency; Higher cost of cloud services	Minimize cloud dependency; Low cost, low power servers closer to end user
Service Model	Integrated service	Reliance on data and intelligence stored in the cloud	Federated
Access	Fixed point access	Limited portability	Portability
	Network dependency	Dependence on high throughput	Network interdependent layers
Machine Learning	ML requires real-time communication with cloud	ML fully dependent on Internet connectivity	Buffering layer that conducts ML without cloud
Data Model	Integrated data models	Centralized data and model management	Data decoupled from model repository
Security	Centralized	Higher vulnerability	Data blind

Table 1. Prevalent Industry Practice and Preferred Approach for FML

The table above compares the prevalent industry practice with the preferred approach for FML across the dimensions of architecture, service model, access, network dependency, machine learning, data model, and security.

A common architecture that is prevalent today is a cloud-client model, where the model training is performed at the cloud based server that are accessed from the end user devices. However, as the architecture is completely cloud dependent it requires consistent Internet connectivity and involve higher associated costs. A low cost low power edge server that serves as a buffer between the end user and the cloud server is preferred as it reduces dependency on the cloud. For the service model dimension, the industry norm is one of an integrated approach. This involves integrated data and models stored in the cloud. A federated system would reduce this dependence, as well as decouple data from the models. For the access dimension, the industry practice is one of a fixed point access resulting in limited portability, and for network dimension, the industry standard is network dependent due to high throughput requirements. ML is typically based on real-time communication with the cloud but a buffering layer could provide model interaction without the need for continuous connectivity with the cloud. Security dimension in the integrated data model is typically a centralized management approach. The drawback is that it is more vulnerable. The proposed way forward is a data blind approach, where the data is anonymized and federated.

A supporting architecture for FML should provision the following capabilities. First, although most users may have ubiquitous access to Internet and data, the architecture should be respectful of this fact, particularly in scenarios where Internet throughput is limited and data access is expensive. Second, the architecture should enable seamless collection and management of privacy-sensitive multi-dimensional dataset from the end users. The data movement to and from end user devices should be in a compressed and granular format. Third, model training and update in the cloud should not require dependence on a continuous stream of data from the end user devices. Forth, the architecture should not assume independently and identically distributed (IID) data. The idea arises from realistic scenarios whereby different type of users interact with the algorithms in different ways, and thus the action of one user may not be interchangeable with another user. Furthermore, there may be more data from one type of user (e.g., a clinical trial participant), and less from another type of user (e.g., a bioinformatics researcher). This imbalance should not skew model training or overfit models for common use cases. On the contrary, the differential between individual use cases and the master model should be anonymized and blurred through an averaging process, while respecting the privacy of the individual user. This reconciliation also ensures that the models are not thrown off by outlier user cases. Fifth, the data analytics pipeline should be extensible and replicable, such that it moves machine learning closer to edge devices and end users. Sixth, federate machine learning should utilize natural labeling of end user data in the model training and knowledge discovery process (here, natural labeling implies the identification of use cases based on user's interaction with applications, thus eliminating the need for human intervention for labeling unique use cases).

FML Architecture

We propose an architecture for FML as shown in Figure 2. The architecture aims to address the preferred directions as stated in the previous sections. It consists of three tiers, namely cloud, low power edge server, and end user devices.

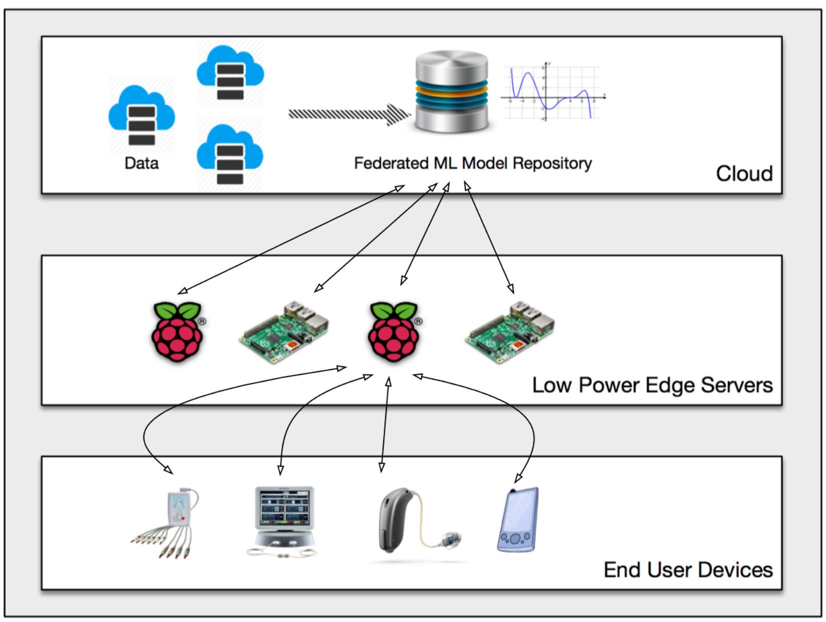

Figure 2. Federated Machine Learning Architecture

At the top is the **cloud** tier, which utilizes the cloud for data storage, model generation, and model management. Anonymized data is stored in the cloud and reconciled models are maintained in the FML model repository. The repository to store the models is built using MongoDB. The cloud layer maintains a clear separation between the FML model repository and the user data. It thus offers enhanced data security by controlling and limiting how the models use the data for training and testing. Furthermore, the models themselves are pushed to the low power edge servers in the middle tier. The users interact with the models on the low power edge servers in the middle tier and have no direct access to the cloud layer. This adds an additional layer of security between the end user and the data stored in the cloud.

The low power edge servers in the middle tier provide affordable computing capabilities. The models are stored at this layer for consumption by the end user applications. Low cost servers make them more attractive for deployment in resource constrained environments such as a healthcare clinic at a rural location. They have the advantage of portability as would be required in an ambulatory setting. Additionally, they reduce dependency on the cloud as only the relevant and application specific models are stored at this layer. End user devices access the models directly from this layer over a localized network, and without the need for Internet connectivity to the cloud. The low power edge servers periodically access the cloud and receive the newly updated model. This approach largely reduces cloud access and data transfer frequency, thereby resulting in moderated cloud dependency.

Google has pioneered the use of advanced deep learning approaches such as Tensorflow, but its execution on the end user devices is possible only on new and powerful mobile devices. This may not be taken for granted. In many instances, TI requires data gathering and ML algorithms to run closer to the end user devices and sensors. The low power edge servers serve as a middle layer to mediate the execution of the computationally intense ML algorithms. It eliminates the need for reliance on ML algorithms stored in the cloud.

End user devices are the primary means by which the subjects interact with the models. They may vary in terms of size, purpose, and utility ranging from mobile devices to smart sensors and wearable technology devices. These devices are registered to the low power edge servers in the middle tier. Data exchange between the end user devices and low power edge servers are encrypted and anonymized. Privacy-sensitive data is moved from the end user device to the middle tier in a compressed and granular format. The middle tier temporarily stores the newly collected user data before it is passed to the cloud tier.

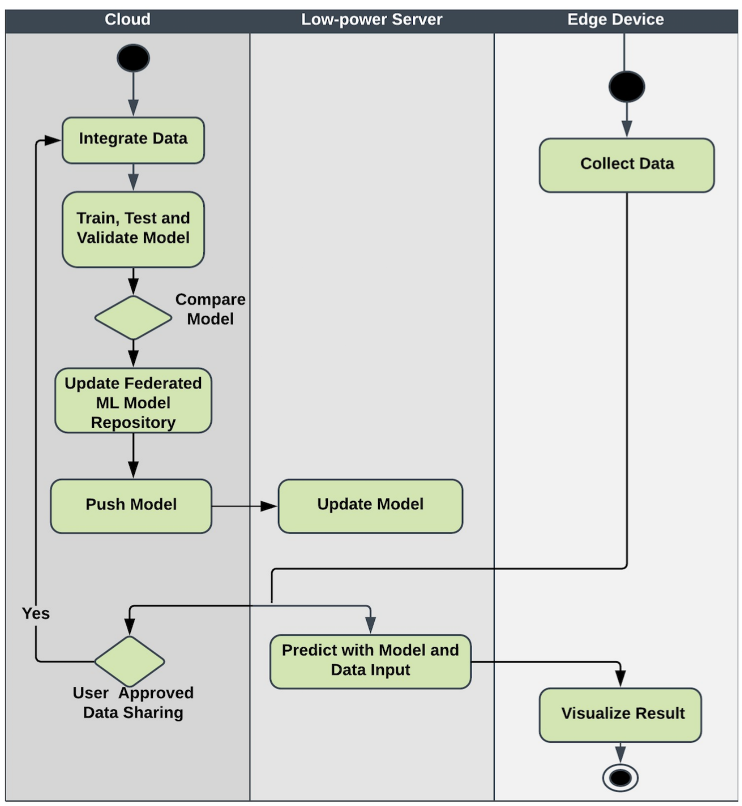

Figure 3. Federated ML workflow

The architecture implicitly provides the ability to federate ML in two distinct ways. First, it accommodates for the difference in the way different users interact with the models. For example, the architecture can determine if more data from one use case may unfavorably skew the model outcome for another use case scenario. Second, the architecture federates the model training and update required over time. Data from the individual user interaction and model feedback are collected, combined, and transferred from the middle tier to the cloud tier. Reconciliation of master models is managed in the cloud tier, averaging out individual variations, while preserving the privacy of the user. Master models are then saved to the FML model repository. The workflow for our proposed FML architecture is depicted in Figure 3.

Evaluation

Table 1 identified five dimensions for the evaluation of FML, namely architecture, service model, access, machine learning, and security. In this research, we present an evaluation of the architecture for its operational performance.

A key difference between our proposed architecture and traditional approaches is the introduction of the low power edge servers in the middle tier. To assess the feasibility of our proposed architecture, we tested three propositions related to operational performances.

P1: *For small dataset, performance of the middle tier (i.e., the low power edge servers) is comparable to high end servers.*

P2: *For large dataset, performance of the middle tier (i.e., the low power edge servers) is comparable to high end servers.*

P3: *For large dataset, performance the middle tier (i.e., the low power edge servers) will be within tolerable and acceptable range.*

Sample Size (n)	**Architecture**	**Training/Prediction Time to prediction**	**Execution Time (ms)**			
			LR	NB	SVM[1]	SVM.025[1]
1000	Federated ML architecture	Training (cloud server[2])	7.982	1.166	207.353	120.115
		Prediction (low power edge server[3])	0.571	1.143	0.947	0.789
	Cloud Server	Prediction (cloud server)	0.548	0.273	0.345	0.204
100000	Federated ML architecture	Training (cloud server[2])	534.727	28.657	269016	108763
		Prediction (low power edge server[3])	0.586	1.139	35.672	10.271
	Cloud Server	Prediction (cloud server)	1.292	0.371	8.281	1.387

1. SVM-Support Vector Classification (1. gamma=2, C=1; and 2. kernel=linear, C=0.025;)
2. Cloud server: Virtual Machine with 2.2 GHz Intel Core i7 processor, 16GB RAM, 1600 MHz DDR3 HD
3. Low power edge server: Raspberry Pi 3 with 1.2 GHz ARM Cortex processor, 1GB RAM, 900 MHz 64Gb SD card

Table 2. Performance of Low Power Edge Server

To conduct a comparative assessment of the operational performance, we used a diabetics dataset consisting of 10 years of clinical care data from 130 US hospitals. The data consists of over 50 features representing patient and hospital outcomes. Model engineers may choose a specific subset of features from the dataset to generate machine learning models based on predetermined clinical decision support requirement. For this evaluation, we selected eight features and one dependent variable, 'readmit.' Since we were interested in model prediction capabilities of the low power edge servers in the middle tier across different size dataset for different ML algorithms, we created two datasets of sizes 1000 and 100,000. We then reviewed the operational performance (i.e., time to prediction) for four ML model prediction methods, namely Logistic regression, Naive Bayes, and two types of Support Vector Machines. We tested two types of SVM (gamma=2, C=1; and kernel=linear, C=0.025;) so that we could determine how the architecture performed with variations in the model parameters. It is to be noted that, in our proposed approach, federated model training is conducted only in the cloud tier. This is similar to current approaches that utilize cloud base ML. What distinguishes our proposed architecture is that model prediction is conducted at the middle tier, which is in contrast to current approaches where the prediction is normally undertaken at the cloud layer. As discussed earlier, the latter requires dependence on data and intelligence stored in the cloud, which may not be taken for granted in TR studies.

For the small sample dataset (n = 1000), our experiments indicated that the model prediction response for the low power edge servers ranged from 0.571ms for LR to 1.143ms for NB. The prediction time for cloud servers ranged from 0.204ms for SVM.025 (kernel=linear, C=0.025) to 0.548ms for LR. For small datasets, the prediction time for LR is comparable for both low power edge server and cloud server. The response times for the remaining ML algorithms are slower for low power edge server. For the larger sample dataset (n = 100000), the low power edge server yielded a wider range of model prediction time. For example, the prediction time for LR was 0.586ms whereas the prediction time for SVM (gamma=2, C=1) was 35.672ms. Therefore, the results do not confirm P1 and P2. To assess whether the operational performance is tolerable and acceptable, we used guidelines from Nah (2004), and set a benchmark response time of 2000ms as a tolerable response time for information retrieval. The experiment results for both small and large sample dataset were well within the acceptable and tolerable response time benchmark of 2000ms, thus confirming P3.

Discussion of findings and implications

Findings

The results of our experiment showed that low power edge servers perform slower in comparison to the operational performance of cloud server. Considering the significantly powerful configuration of the cloud server this may not come as a surprise. Nevertheless, it is interesting to note that the low power edge servers performed well within the acceptable and tolerable levels for information retrieval resulting from model predictions. The ML algorithms at the middle tier were implemented using Miniconda[1]. As a scaled down version of Anaconda, Miniconda provides support for only a subset of ML algorithms. In addition to the ML algorithms that we presented in Table 2, other popular ML algorithms supported by Miniconda include Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Decision Trees (DT), and Neutral network. We do not include the evaluation of these models in our research due to space limitations.

Our experiments also indicated that all ML algorithms we tested out-performed our operational benchmark (Nah, 2004). Thus, it is fair to conclude that end users will not be affected by the marginally lower performance of the middle tier. Our study showed that low power edge servers in our proposed design equivalent prediction capabilities as that of cloud-based service at a much lower cost (the cost of a Raspberry Pi server is $30 compared to the $370 monthly data transfer cost of a cloud-based solution used in our experiment). Thus, we argue that the benefits of FML far outweighs the limitations of the low power edge servers.

Practical and theoretical contributions

Sepsis is defined as a life threatening organ dysfunction caused by an uncontrolled host response to infection (Hotchkiss et al., 2016). Efforts to use machine learning to predict sepsis in the Intensive Care Unit have achieved positive results (Desautels et al., 2016). However, they rely mostly on vital data reported in Electronic Health Records (EHR). Variables typically used for developing ML models for the assessment of adult sepsis include age, sex, acuity, blood pressure, pulse rate, pain scale, and temperature. The vitals currently used in ML models are not collected in real time constant, but gathered through periodic human interventions and recorded in EHR. The discrete data is thus limiting for real time monitoring and prediction of sepsis. The FML architecture is well positioned to address these challenges by collecting data from different sources (e.g., wearable and proximity sensors), and providing clinical decision support in a network agnostic manner.

The solution is portable to various settings such as a hospital triage, ambulatory clinics, and even the patient's home. The approach for FML leverages low power edge servers to mediate the communication between cloud and the human agency. A low power server is housed at the sepsis assessment location. A patient wearing sensors generates streaming data which is run against the sepsis prediction model stored

[1] http://repo.continuum.io/miniconda/Miniconda3-latest-Linux-armv7l.sh

on the low power server. The prediction outcome is presented to the human agent for clinical decision support. If the predictions for a particular patient is determined to be inaccurate beyond a threshold, it is an indication that there is an imbalance in the data. Data from that specific low power server is then sent to the cloud to retrain the model. Data from different individuals are collected, combined, and averaged across all similar use-cases. The difference between the prediction for individuals and the master model are aggregated to train and update the models in the cloud. Changes associated with each individual is anonymized and blurred through the averaging process, while respecting the privacy of individual user. The reconciliation ensures that the sepsis prediction model is not thrown off by outlier use-cases.

Although socio-technical perspectives are emphasized in TR (Payne et al., 2009; Sung et al., 2003), a review of related literature indicates that it has been mostly left unaddressed. A plausible explanation is the lack of expertise essential to integrate emergent technologies and socio-technical perspective in TR. Our study illustrates how IS scholarship may inform and fill this crucial gap. Specifically, we demonstrated how IS scholarship related to machine learning and data analytics were brought together to benefit TI through FML. In this study, we demonstrated how FML architecture may be used in TI. The proposed architecture can be adapted to other research areas targeting knowledge management and big data analytics.

Limitations and Future directions

It could be argued that the middle tier introduces an added level of system complexity. It may seem that employing low power edge servers may negatively impact performance and scalability of the data analytics pipeline. However, integrating new technologies such as micro-services (Thönes, 2015) address these concerns. We limit our discussion on this topic due to space considerations, and set aside related discussion for future studies. Another limitation of our study is that we used existing Diabetics dataset for our evaluation. We did not use end user data in our experiment. Future research will utilize end user data for a more comprehensive assessment of the federate ML architecture.

Conclusion

Although big data analytics and machine learning are driving digital transformation across all industries, TI has lagged in its adoption and use. Knowledge management challenges in TI include imbalance of data, privacy and confidentiality concerns, proprietary data distributed across multiple end user devices, and extensive reliance on cloud. In this research, we propose a federated machine learning platform that leverages low power servers to mediate the execution of the ML algorithms. The proposed approach aims to move machine learning closer to edge devices and end users. Positioning machine learning closer to the end-user enables the human agency to interact with ML models without constant access to massive data and intelligence stored in the cloud. Applications of TI in areas such as clinical trials and clinical practice can greatly benefit from the scalability, cost effectiveness, and improved quality of decision support from the FML architecture.

References

Baldi, P., & Long, A. D. (2001). A Bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes. *Bioinformatics, 17*(6), 509-519.

Beaulah, S. A., Correll, M. A., Munro, R. E., & Sheldon, J. G. (2008). Addressing informatics challenges in Translational Research with workflow technology. *Drug discovery today, 13*(17-18), 771-777.

Beck, J. M., Ma, W. J., Kiani, R., Hanks, T., Churchland, A. K., Roitman, J., . . . Pouget, A. (2008). Probabilistic population codes for Bayesian decision making. *Neuron, 60*(6), 1142-1152.

Becker, K. G., Hosack, D. A., Dennis, G., Lempicki, R. A., Bright, T. J., Cheadle, C., & Engel, J. (2003). PubMatrix: a tool for multiplex literature mining. *BMC bioinformatics, 4*(1), 61.

Castaneda, C., Nalley, K., Mannion, C., Bhattacharyya, P., Blake, P., Pecora, A., . . . Suh, K. S. (2015). Clinical decision support systems for improving diagnostic accuracy and achieving precision medicine. *Journal of clinical bioinformatics, 5*(1), 4.

Chen, J., Qian, F., Yan, W., & Shen, B. (2013). Translational biomedical informatics in the cloud: present and future. *BioMed research international, 2013*.

Coulouris, G. F., Dollimore, J., & Kindberg, T. (2005). *Distributed systems: concepts and design*: pearson education.

Desautels, T., Calvert, J., Hoffman, J., Jay, M., Kerem, Y., Shieh, L., . . . Barton, C. (2016). Prediction of sepsis in the intensive care unit with minimal electronic health record data: a machine learning approach. *JMIR medical informatics, 4*(3).

Fielding, R. T., & Taylor, R. N. (2000). *Architectural styles and the design of network-based software architectures* (Vol. 7): University of California, Irvine Doctoral dissertation.

Fontanarosa, P. B., & DeAngelis, C. D. (2002). Basic science and translational research in JAMA. *Jama, 287*(13), 1728-1728.

Gholami, R., Watson, R. T., Molla, A., Hasan, H., & Bjørn-Andersen, N. (2016). Information systems solutions for environmental sustainability: How can we do more? *Journal of the Association for Information Systems, 17*(8), 521.

Harris, P. A., Taylor, R., Thielke, R., Payne, J., Gonzalez, N., & Conde, J. G. (2009). Research electronic data capture (REDCap) - A metadata-driven methodology and workflow process for providing translational research informatics support. *Journal of Biomedical Informatics, 42*(2), 377-381.

Holzinger, A. (2016). Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics, 3*(2), 119-131.

Hotchkiss, R. S., Moldawer, L. L., Opal, S. M., Reinhart, K., Turnbull, I. R., & Vincent, J.-L. (2016). Sepsis and septic shock. *Nature Reviews Disease Primers, 2*, 16045.

Koski, L. B., Gray, M. W., Lang, B. F., & Burger, G. (2005). AutoFACT: An auto matic functional annotation and classification tool. *BMC Bioinformatics, 6*(1), 151.

Kuderer, M., Gulati, S., & Burgard, W. (2015). *Learning driving styles for autonomous vehicles from demonstration*. Paper presented at the Robotics and Automation (ICRA), 2015 IEEE International Conference on.

Lee, J. J., Hong, W. K., Hittelman, W. N., Mao, L., Lotan, R., Shin, D. M., . . . Papadimitrakopoulou, V. M. (2000). Predicting cancer development in oral leukoplakia: ten years of translational research. *Clinical Cancer Research, 6*(5), 1702-1710.

Li, Y., Thomas, M. A., & Osei-Bryson, K.-M. (2017). Ontology-based data mining model management for self-service knowledge discovery. *Information Systems Frontiers, 19*(4), 925-943.

Lindgren, H. (2008). Decision support system supporting clinical reasoning process-an evaluation study in dementia care. *Studies in Health Technology and Informatics, 136*, 315.

McMahan, B., & Ramage, D. (2017). Federated Learning: Collaborative Machine Learning without Centralized Training Data. Retrieved from https://research.googleblog.com/2017/04/federated-learning-collaborative.html?m=1

Nah, F. F.-H. (2004). A study on tolerable waiting time: how long are web users willing to wait? *Behaviour & Information Technology, 23*(3), 153-163.

Newman, S. (2015). *Building microservices: designing fine-grained systems*: " O'Reilly Media, Inc.".

O'neill, E. S., Dluhy, N. M., & Chin, E. (2005). Modelling novice clinical reasoning for a computerized decision support system. *Journal of Advanced Nursing, 49*(1), 68-77.

Pan, W. (2002). A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments. *Bioinformatics, 18*(4), 546-554.

Payne, P. R., Embi, P. J., & Sen, C. K. (2009). Translational informatics: enabling high-throughput research paradigms. *Physiological genomics, 39*(3), 131-140.

Porollo, A. A., Adamczak, R., & Meller, J. (2004). POLYVIEW: A flexible visualization tool for structural and functional annotations of proteins. *Bioinformatics, 20*(15), 2460-2462.

Rubio, D. M., Schoenbaum, E. E., Lee, L. S., Schteingart, D. E., Marantz, P. R., Anderson, K. E., . . . Esposito, K. (2010). Defining translational research: implications for training. *Academic medicine: journal of the Association of American Medical Colleges, 85*(3), 470.

Singleton, A. (2016). The Economics of Microservices. *IEEE Cloud Computing, 3*(5), 16-20.

Sung, N. S., Crowley Jr, W. F., Genel, M., Salber, P., Sandy, L., Sherwood, L. M., . . . Getz, K. (2003). Central challenges facing the national clinical research enterprise. *Jama, 289*(10), 1278-1287.

Sussman, S., Valente, T. W., Rohrbach, L. A., Skara, S., & Ann Pentz, M. (2006). Translation in the health professions: Converting science into action. *Evaluation & the Health Professions, 29*(1), 7-32.

Thönes, J. (2015). Microservices. *IEEE Software, 32*(1), 116-116.

Westfall, J. M., Mold, J., & Fagnan, L. (2007). Practice-based research: "Blue Highways" on the NIH roadmap. *Jama, 297*(4), 403-406.

Woolf, S. H. (2008). The meaning of translational research and why it matters. *Jama, 299*(2), 211-213.