**Grant Number: 1839007**

**Project Title: EAGER: REAL-D: Smart Decision Making using Data**

**PD/PI Name: Marianthi Ierapetritou, Principal Investigator**

**Shantenu Jha, Co-Principal Investigator**

**Rohit Ramachandran, Co-Principal Investigator**

**Reporting Period: 09/01/2018 - 08/31/2019**

The main aim of this project is to scientifically and technologically advance the operability of a continuous and integrated pharmaceutical process. This can be accomplished by enabling real-time learning and decision-making for continuous process improvement and risk mitigation. The work attempts to bridge the gaps in manufacturing data integration, laboratory data management, and real-time decision making for continuous pharmaceutical manufacturing processes. The work involves the following major steps:

* Development of a data integration framework to enable data management for plant and laboratory data
* Identify process knowledge extraction strategies to enable the use of collected data for process improvements and model development

Main results:

In the past year, we have made progress on the following fronts with respect to specific deliverables:

1. General development of data integration framework
2. Integration of material property and analytical data into the cloud
3. Integration of continuous manufacturing plant data into data historian and cloud
4. Integration of process analytical tools (PAT) and process equipment to control software and data historian
5. Development of a case study to test data-driven modeling approaches for process prediction of the continuous pharmaceutical manufacturing processes
6. Development of data transfer reference architecture for utilizing data with high performance computing (HPC) resource
7. **Development of a data integration framework**

With the recent advancement in computing, communications devices and technology has increased research attention towards Internet of Things (IoT) in automation of manufacturing (Liao, Deschamps et al. 2017). Cyber-physical systems (CPS) are also gaining importance to increase automation using several computer-based algorithms (Zhong, Xu et al. 2017). This has led several regulatory agencies to push forward an IoT framework to help increase output quality (CDER 2017). Along with the QbD, PAT, and RTRT initiatives, the amount of data that can be collected from the pharmaceutical manufacturing process increases dramatically. To fully exploit the potential of the available data, an integrated data acquisition, storage, integration, and knowledge extraction process is in need.

A pilot for data integration is being developed at Rutgers University to test an IoT framework for continuous manufacturing (CM) of pharmaceutical products. Integration of process data from the continuous manufacturing plant with analytical data obtained from the laboratory tests can follow ISA-88 standards (Cao, Mushnoori et al. 2018), and can be broken down into several small integration steps. Analytical tests for material properties are usually standalone tests, where data collection is discrete and does not require continuous data acquisition. On the other hand, continuous acquisition is required from various equipment and sources for the CM plant data. Different strategies need to be implemented to collect different type of data, which could eventually be transferred to a data lake. Thus, the integration methods can be segregated into three broad techniques:

1. Material property and analytical data collection and integration
2. Continuous manufacturing plant data collection and integration
3. Connecting both data to a common data lake

The development of each piece is detailed in the following sections.

1. **Integration of material property / analytical data**

Several experiments need to be performed to obtain material properties for a pharmaceutical product. This data acquired needs to be structured such that it can be extracted into the data. Thus, the data collection needs to be standardized to maintain a standard data structure inside the database. Keeping these two important features in mind, several online solutions for electronic laboratory notebooks (ELNs) were evaluated. Table 1 shows the comparison of several solutions that were used before deciding on the final ELN solution.

After careful evaluation, SciCord was chosen as the ELN solution that would help standardize data collection using its Microsoft Excel-based user interface, which is easy for everyone to understand. Data collection was standardized using templates which were developed in MS Excel. The templates can be made more flexible by automating function with a .XML configuration file. There are several inbuilt functions that help with data management and plotting as well. Several MS Excel functions are also supported within the working sheet of SciCord. It is hosted on a separate instance of Microsoft Azure for the Rutgers University.

**Table 1.** Evaluation of main features of electronic laboratory notebooks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Product | Programming language | Server location | Data analysis | Security Compliance | Current Users |
| Scinote | Ruby on Rails | Scinote cloud or personal server | No tools available | 21CFR part 11 (Premium version) | 10+ Universities |
| Jupyter | Python | Personal server | Can be done on the fly | 21 CFR part 11 needs to implemented manually | Netflix, various data analytics companies |
| SciCord | .NET and C# | Scicord’s cloud / personal server | Basic tools | 21CFR part 11 | Catalent, Aurobindo, AstraZeneca |
| Labstep | Python | Labstep’s cloud | No tools available | 21CFR part 11 | Fairly new, niche case studies available |

**2.1 Salient feature of SciCord**

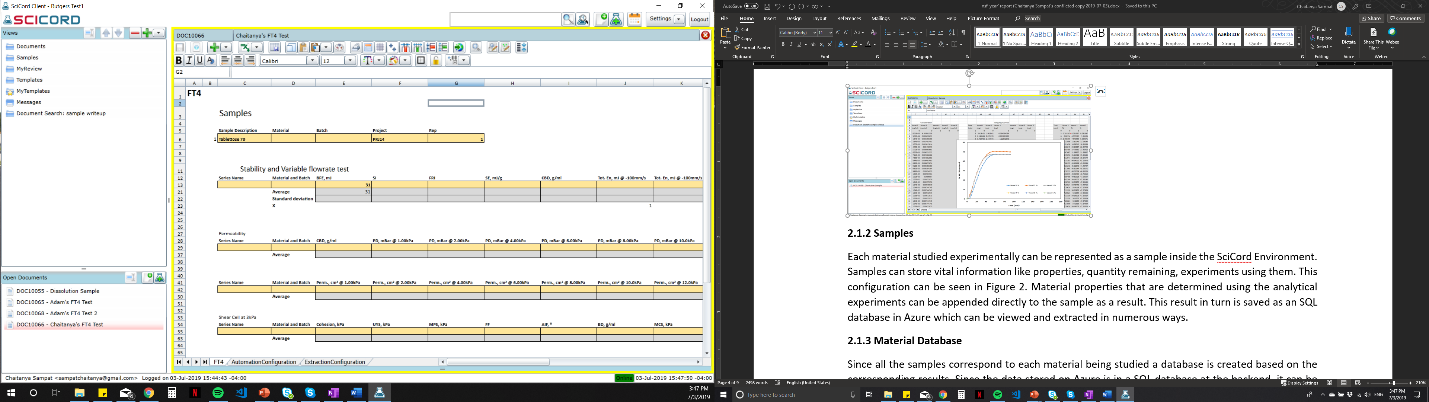
**2.1.1 Template**

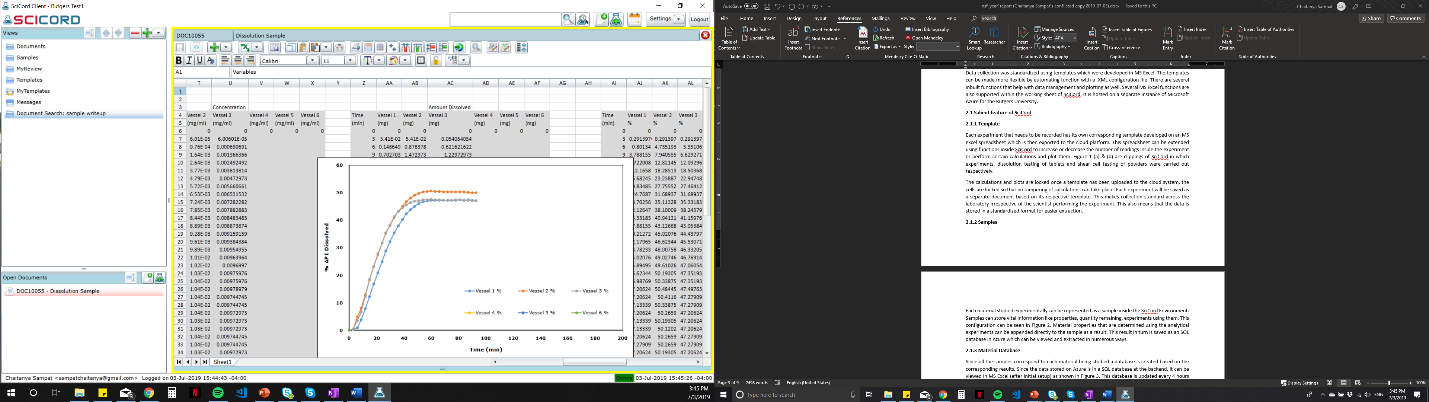
Each experiment that needs to be recorded has its own corresponding template developed on an MS Excel spreadsheet, which is then exported to the cloud platform. This spreadsheet can be extended using functions inside SciCord to increase or decrease the number of readings inside the experiment or perform certain calculations and plot them. Figure 1 (a) & (b) are clippings of SciCord in which experiments, dissolution testing of tablets and shear cell testing of powders were carried out respectively.

The calculations and plots are locked once a template has been uploaded to the cloud system. The cells are locked so that no tampering of calculations can take place. Each experiment will be saved as a separate document based on its respective template. This makes collection standard across the laboratory irrespective of the scientist performing the experiment. This also means that the data is stored in a standardized format for easier extraction of data.

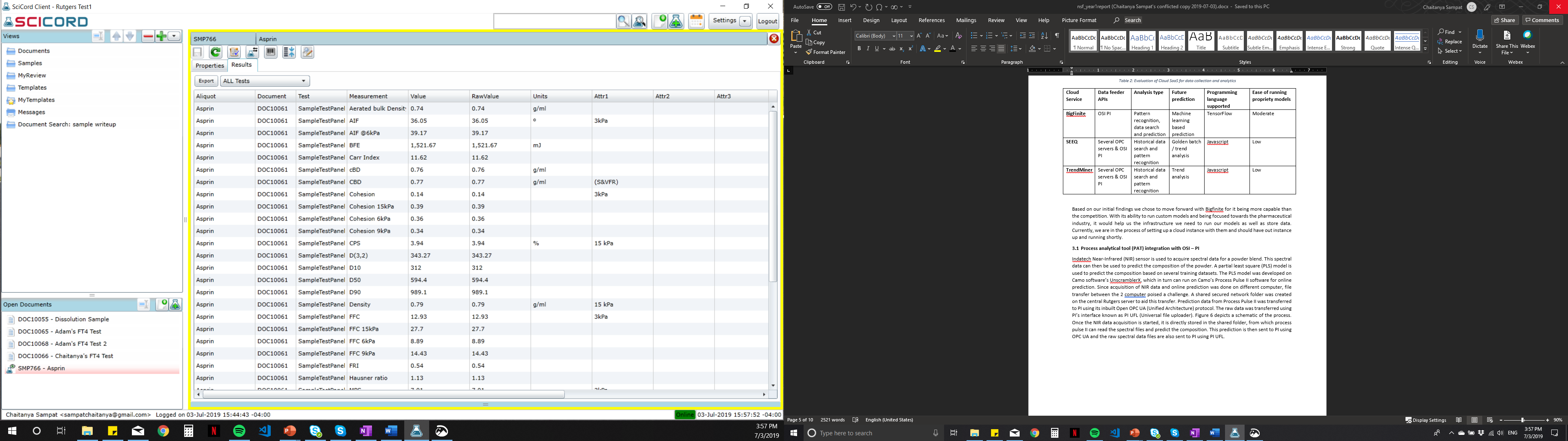
**2.1.2 Samples**

Each material studied experimentally can be represented as a sample inside the SciCord Environment. Samples can store vital information like properties, quantity remaining, experiments using them. This configuration can be seen in Figure 2. Material properties that are determined using the analytical experiments can be appended directly to the sample as a result. This result in turn is saved as an SQL database in Azure which can be viewed and extracted in numerous ways.





**Figure 1.** Templates for data collection developed in SciCord for (a) Tablet dissolution testing and for (b) shear cell and permeability testing.



**Figure 2.** Material properties stored as results inside samples.

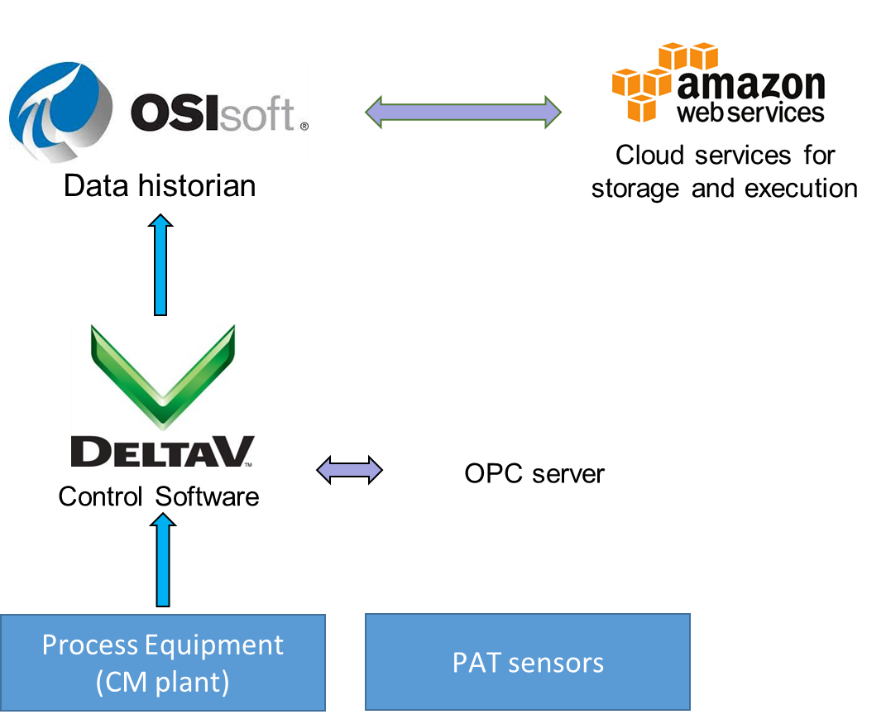
**2.1.3 Material Database**

Since all the samples correspond to each material being studied a database is created based on the corresponding results. Since the data stored on Azure is in a SQL database at the backend, it can be viewed in MS Excel (after initial setup). This database is updated every 4 hours so the data being pulled is always up to date. Since models in this study will run on high performance computing (HPC) systems, automated data extraction can be performed using Python.

1. **Integration of CM plant data**

In contrast to the analytical data collection, the continuous data acquisition of the CM plant poses its own challenges. The running of the plant is dependent on the recipes that are developed within the control software which can acquiesce data from the OLE (Object Linking and Embedding) for Process Control (OPC) interface of the equipment and sensors. The control software does not have the capabilities to save the data for an extended period. Thus, there is a need for a data historian to store the data. This has been achieved in the CM plant at Rutgers by using OSI-PI. Figure 3 shows an example workflow of data in the continuous plant being used.

This CM data also needs to be sent to a data lake to store data and access it when required. Several software as service (SaaS) were evaluated based on their ability to store data, perform data analytics etc. Table 2 shows a comparison between 3 SaaS we evaluated.



**Figure 3.** Schematic of data flow for the CM plant

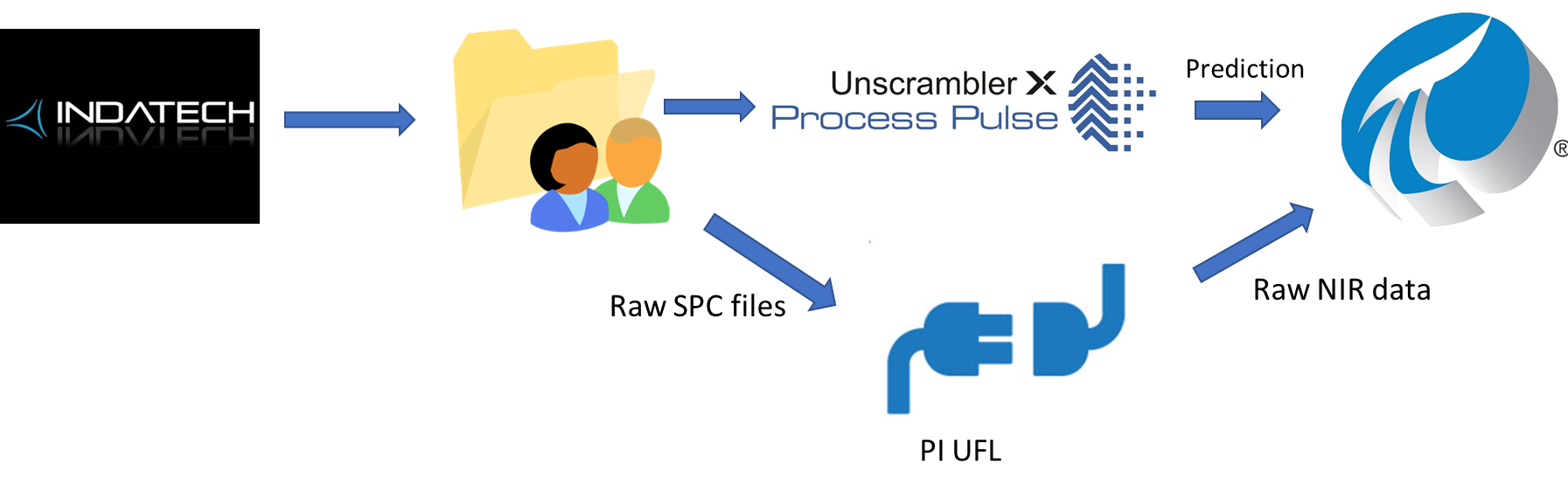
**Table 2.** Evaluation of Cloud SaaS for data collection and analytics

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cloud Service** | **Data feeder APIs** | **Analysis type** | **Future prediction** | **Programming language supported** | **Ease of running propriety models** |
| **BigFinite** | OSI PI | Pattern recognition, data search and prediction | Machine learning based prediction | TensorFlow | Moderate |
| **SEEQ** | Several OPC servers & OSI PI | Historical data search and pattern recognition | Golden batch / trend analysis | Javascript | Low |
| **TrendMiner** | Several OPC servers & OSI PI | Historical data search and pattern recognition | Trend analysis | Javascript | Low |

Based on our initial findings, we chose to move forward with Bigfinite for it being more capable than the competition. With its ability to run custom models and being focused towards the pharmaceutical industry, it would help us the infrastructure we need to run our models as well as store data. Currently, we are in the process of setting up a cloud instance with them and should have out instance up and running shortly.

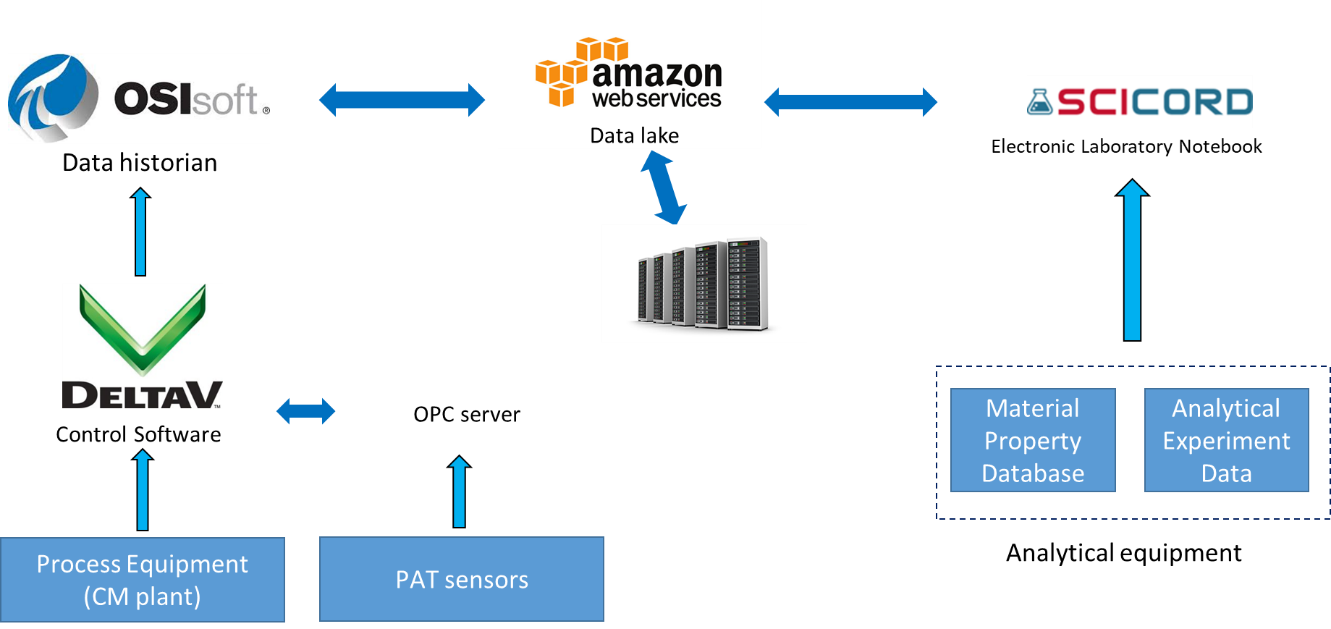
1. **Process analytical tool (PAT) integration with OSI – PI**

Indatech Near-Infrared (NIR) sensor is used to acquire spectral data for a powder blend. This spectral data can then be used to predict the composition of the powder. A partial least square (PLS) model is used to predict the composition based on several training datasets. The PLS model was developed on Camo software’s UnscramblerX, which in turn can run on Camo’s Process Pulse II software for online prediction. Since the acquisition of NIR data and the online prediction was done on different computers, file transfer between the 2 computer poised a challenge. A shared secured network folder was created on the central Rutgers server to aid this transfer. Prediction data from Process Pulse II was transferred to PI using its inbuilt Open OPC UA (Unified Architecture) protocol. The raw data was transferred using PI’s interface, known as PI UFL (Universal file uploader). Figure 4 depicts a schematic of the process. Once the NIR data acquisition is started, it is directly stored in the shared folder, from which process pulse II can read the spectral files and predict the composition. This prediction is then sent to PI using OPC UA and the raw spectral data files are also sent to PI using PI UFL.



**Figure 4.** Schematic of process flow of NIR data to OSI-PI

Combining all aforementioned data integration work, a proposed framework for the complete integration of data from the CM line and analytical laboratory is shown in Figure 5.



**Figure 5.** Complete data integration framework for CM plant and analytical laboratory data

1. **Data-driven modeling case study on a feeder-refill system**

The data integration framework allows manufacturing data from the plant and analytical laboratories to be collected and stored at a central platform. The large volume of data available gives rises to opportunities in training data-driven models to predict system behaviors. To better understand the capability of machine learning in modeling complex systems, a case study was conducted with a feeder-refill system of the continuous direct compaction line.

The feeder-refill system used in this case study is shown in Figure 6. It consists of a feeder unit, a refill unit, and a PID control loop. Both units contain the same powder, and the feeder unit can output the powder at the rate that is set on the controller. When the weight of powder remaining in the feeder is less than a certain percentage, powder in the refill unit can be transferred to the feeder in order to maintain continuous operation.

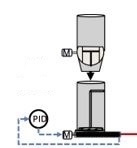
The mechanistic models developed from previous projects (Wang, Escotet-Espinoza et al. 2017) have identified that design and process parameters of the system, as well as material properties of the powder in the system, are all important in determining the mass flowrate output. Among all these variables, the design parameters are usually fixed when one develops the system, and are not able to change easily. Therefore, in this case, the impacts of process parameters and material properties onto the mass flowrate output were explored. Nine variables were identified, and the mass flowrates out of the feeder were simulated with these nine varying parameters from 0 to 100 seconds. All variables were set within their corresponding boundaries, and a three-level full factorial design was carried out. The list of variables and their levels are summarized in Table 3.

Refill

Material Properties

Process Parameters

Design Parameters



Feeder

Mass Flowrate Out

**Figure 6.** Schematic of the feeder-refill system

**Table 3.** Process variables, material properties and their set levels

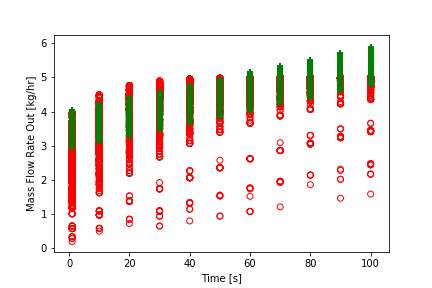
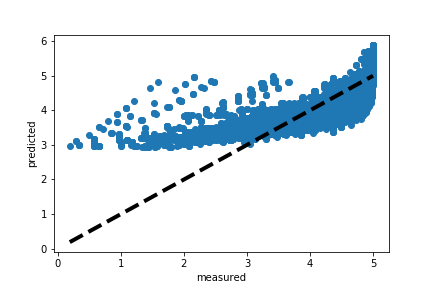
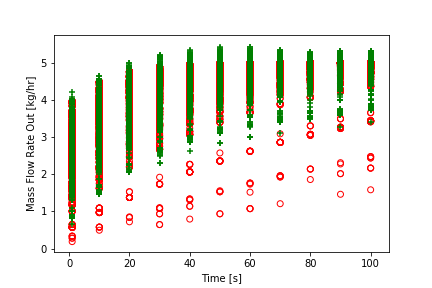
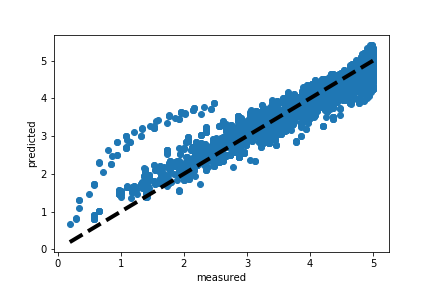
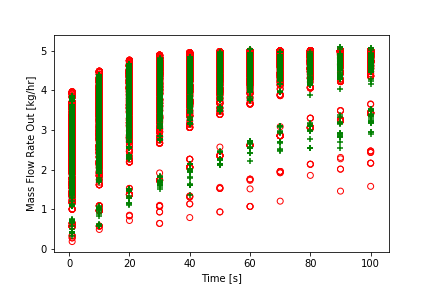
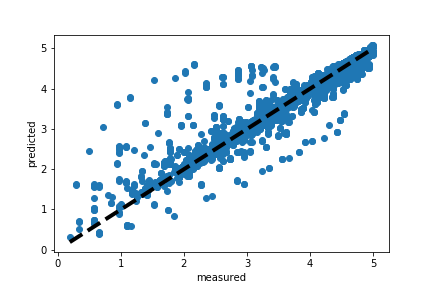
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Factors** | **Units** | **Low** | **Medium** | **High** |
| Flowrate set point | kg/h | 5 | 10 | 15 |
| Refill on set fill level | fraction | 0.3 | 0.5 | 0.7 |
| Initial mass in feeder | kg | 0.7 | 1.0 | 1.2 |
| Bulk density | kg/m3 | 300 | 450 | 650 |
| Angle of friction | Degree | 15 | 25 | 35 |
| Cohesion value | kPa | 0.2 | 0.6 | 1.0 |
| Compressibility | % | 5 | 25 | 40 |
| Flow function | -- | 1 | 5 | 20 |
| Permeability | cm2 | 10 | 50 | 150 |

With the simulated data of this input-output correspondence, data-driven regression methods were used to investigate the capabilities of these data-driven models. In this case study, lasso regression, polynomial regression with lasso, and support vector regression were used to model the system. Both lasso regression and polynomial regression with lasso belonged to the large category of variable subset selection (also known as feature selection) method. The variable subset selection method was implemented in this case because the system has a large number of features, and the method could help reduce the number of features and the complexity of the model. With this method, the regressions took the form of least square regression with an added regularization term to penalize coefficient of features, as shown in Equation 1. The support vector regression, on the other hand, developed a hyperplane that maximizes margins between the points to minimize error under certain tolerance (Smola and Scholkopf 2004).

|  |  |
| --- | --- |
|  | (1) |

Training dataset from the simulated data (80% of the full dataset) was used to train the data-driven models, and the performance of all models was assessed with the testing dataset (20% of the full dataset). Their performances are shown in Figure 7 and Table 4.

From Figure 7, it can be concluded that the system behavior is highly non-linear. The polynomial and support vector regression perform much better. In addition, it is obvious that both polynomial regression and support vector regression are able to predict points around the set point much better than points that are much lower than the setpoint. With further examination, it was found that the number of data points that were on the low end only makes up less than 0.2% of the entire dataset. With this small proportion of data, not much weight was put onto these points in the data-driven models, leading to the failure to predict these outputs. In addition, the points with low mass flowrate output were associated with the combination of the following variables: bulk density of 300 kg/m3, permeability of 150 cm2 and compressibility of 5%. No powder from our material property database has properties that are close to this set of combination. Whether or not this combination would result in an experimental output behavior like what is shown in the figure is still a question. If the experimental behavior differs, it may be an indication that the mechanistic model used in this case is not tuned to simulate this set of extreme values. The general observation from this case also provides a significant point that one cannot blindly put data into data-driven models and use them to predict system behaviors as domain expertise on datasets, systems being modeled and models are critically important, and all of them will have an impact on the overall performance of the predictive models.



Linear Lasso

Set Point=5

Polynomial

Set Point=5

SVR-RBF

Set Point=5

**Figure 7.** Performance of 3 data-driven models at flowrate set point of 5 kg/h. Plots on the left depict predicted vs. measured values, and points on the diagonal line indicate a perfect match. Plots on the right shows the time series change of mass flowrate out; red circles represent actual values given by simulations and green marks represent predicted values given by the data-driven model.

**Table 4.** Performance of the three data-driven models at different controller set points

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Data-Driven Model** | **R2** | **RMSE** |
| Set Point = 5 kg/h | Lasso (Linear) | 0.6043 | 0.5291 |
| Polynomial with Lasso | 0.9341 | 0.2159 |
| SVR-RBF | 0.9484 | 0.1911 |
| Set Point = 10 kg/h | Lasso (Linear) | 0.5956 | 1.0744 |
| Polynomial with Lasso | 0.9293 | 0.4492 |
| SVR-RBF | 0.9350 | 0.4308 |
| Set Point = 20 kg/h | Lasso (Linear) | 0.5414 | 2.3915 |
| Polynomial with Lasso | 0.9019 | 1.1062 |
| SVR-RBF | 0.8783 | 1.2320 |

Future work on the exploration of data-driven models will expand the case to the entire continuous pharmaceutical manufacturing line, and investigate on more machine learning approaches such as neural networks and deep learning. With the increase in complexity, high performance computing resources will be needed to train and validate the data-driven models, which is also a focus of this project.

1. **HPC-enabled machine learning for real-time process improvement**

As illustrated in previous sections, an integrated manufacturing data stream can provide a large volume of data, and in order to advance smart decision-making process, data collected need to be analyzed in real-time. Under this idea, typical workstation resource is not suitable for the task as the computational resource is limited. High performance computing resource is then needed to enable real-time analysis.

Before utilizing HPC resource, the main components of the system need to be identified, and a reference architecture needs to be developed to guide the information flow. With the continuous pharmaceutical manufacturing pilot plant, three main components are identified that interact and support the use case. The components are: (i) a database, (ii) a workstation, and (iii) a high performance computing resource (HPC). With these three components, a reference architecture, along with the interaction between the components is developed and shown in Figure 8.

The sensor is the main source that gathers data. The recorded data are transferred to a database, where the database stores data. Here, the assumption is that the database is persistent and has unlimited storage space. The workstation acts as a key interactive component where the user queries the database (Arrow 1 and 2) and launches jobs to either train models or perform predictions. To utilize HPC for executing machine learning algorithms at scale, data are transmitted to the HPC (Arrow 3), and results will be reported back to the workstation (Arrow 4).

A screenshot of a cell phone

Description automatically generated with medium confidence

**Figure 8.** Reference architecture. 1) The workstation queries a database for data. 2) Workstation receives data. 3) Data Transmission to HPC and workflow execution. 4) Model or predicted values reported to the workstation.

The general information flow follows this reference architecture. Currently, the connection between sensor and database is built, but the connection between database and workstation (the query mechanism) is yet to be developed. We are utilizing a dedicated public repository for the purposes of development (Real D Github).

Future work will establish a complete architecture, to investigate implementation details (time, accuracy, scalability, suitable learning framework etc.) of multiple machine learning models that are relevant to the use case. In addition, we are performing research on distributed machine learning infrastructure to compare real-time learning and decision-making to batch learning. Real-time learning requires data processing as soon as they are produced and become available. Frameworks with streaming capabilities provide the necessary abstractions and infrastructure to support real-time data processing. To this extend, we plan to utilize Pilot-Streaming (Luckow A. et al (2018)). Pilot-Streaming provides the ability to on-demand deploy and manage streaming framework and their applications. Using pilot streaming will give the advantage of dynamically add/remove resources to a streaming cluster.

**List of Publications and Presentations from this work:**

1. Chen, Y., Dias, L., Metta, N., Ierapetritou, M. G. (2019). Data-driven Modeling of Unit Operations in Continuous Pharmaceutical Manufacturing Line under the Industry 4.0 Framework. *Poster presentation at Machine Learning in Science and Engineering, Atlanta, GA, USA*, June 10-12, 2019.

2. Chen, Y., Ierapetritou, M. G. (2019). Development of Data-Driven and Hybrid Models for Continuous Pharmaceutical Manufacturing Lines Under Industry 4.0 Framework. *Oral presentation at AIChE, Orlando, FL, USA*, November 10-15, 2019.

**References:**

Cao, H., et al. (2018). "A Systematic Framework for Data Management and Integration in a Continuous Pharmaceutical Manufacturing Processing Line." Processes **6**(5).

CDER, F. (2017). "Advancement of Emerging Technology Applications for Pharmaceutical Innovation and Modernization Guidance for Industry."

Liao, Y., et al. (2017). "Past, present and future of Industry 4.0 - a systematic literature review and research agenda proposal." International Journal of Production Research **55**(12): 3609-3629.

Wang, Z., et al. (2017). "Process analysis and optimization of continuous pharmaceutical manufacturing using flowsheet models." Computers & Chemical Engineering **107**: 77-91.

Zhong, R. Y., et al. (2017). "Intelligent Manufacturing in the Context of Industry 4.0: A Review." Engineering **3**(5): 616-630.

Smola, A. and B. Scholkopf (2004). “A tutorial on support vector regression.” Statistics and Computing **14**:199-222.

Real D Github Repo: <http://github.com/radical-collaborations/reald>

Luckow, A. et al. (2018). “Pilot-Streaming: A stream processing framework for High-Performance Computing”, Arxiv.