Using a data lake in animal sciences

D. Schokker[1*], I. N. Athanasiadis[2], B. Visser[3], R. F. Veerkamp[1] and C. Kamphuis[1]
[1]*Wageningen Livestock Research, Wageningen University & Research, 6700AH Wageningen, The Netherlands*
[2]*Information Technology Group, Wageningen University & Research, 6706KN Wageningen, The Netherlands*
[3]*Hendrix-Genetics, 5831CK Boxmeer, The Netherlands*

* dirkjan.schokker@wur.nl

Abstract

In the livestock domain, Big Data is becoming more common and is being anchored into the mind-set of researchers. With the increasing availability of large amounts of data of varying nature, there is the challenge how to store, combine, and analyze these data efficiently. With this study, we explored the possibility of using a data lake for storing and analyzing sensor data, using an animal experiment as use case, to improve scalability and interoperability. The use case was an experiment within Breed4Food (a public-private partnership), in which the gait score of 200 turkeys was determined. In the experiment, a gait score was traditionally assigned to each animal by a highly-skilled person who visually inspected them walking. Next to it, a set of sensor data streams was recorded for each animal, specifically inertial measurement units (IMUs), a 3D-video camera, and a force plate, with the ambition to explore the effectiveness of these data streams as predictors for estimating the gait score. The resulting sensor output, i.e. raw data, were successfully stored in its original format in the data lake. Subsequently, for each sensor output we performed extract, transform, and load activities, by executing custom-made scripts to generate tab or comma separated files. Lastly, by using Apache Spark it was possible to easily perform parallel processing of the data, allowing for fast computing. In conclusion, we managed to set up a data lake, load animal experimental data and run preliminary analyses. The data lake allowed for easy scale up of both data loading and analyses, which is desired for dynamic analyses pipelines, especially when more data are collected in the future.

Keywords: data lake, sensor data, animal experiment, scalability

Introduction

Traditionally, animal scientists work with structured relational databases to store data used in their research. However, ongoing technological innovations and their implementation results in a whole generation of unstructured non-relational data, i.e. camera or video images, and the quantity of data is increasing simultaneously too. With the increasing availability of large amounts of data with varying nature, there is the challenge how to store, combine, and analyse these data efficiently. In the information computation technology world there seems to be a move from structured relational databases to schema-less databases (commonly referred to as NoSQL databases) for management and storage of data; and from data warehouses to data lakes. The key driver behind this transition is that we need to store the source data and handle ever increasing datasets, varying data structures, as well as heterogeneous and multimodal data. To this end, the process of data pre-processing, i.e. the extract, transform, and load (ETL) procedure becomes more demanding.

With this study, we explored the possibility of using a data lake for storing and analysing sensor data, using an animal experiment as use case, to have improved scalability and interoperability. In this experiment, three different sensors were used to capture (indices) of the gait score of turkeys. The three sensors used were 1) inertial measurement units (IMUs), 2) a 3D-video camera, and 3) a force plate. The raw output of these three sensors were stored in the data lake and the ETL procedure was applied by employing custom scripts to make the data more aligned with FAIR-principles.

Material and methods

Data lake

The deployment of a data lake involves the installation and management of several big data software tools, including reliable distributed file systems, cluster resource management and execution environments for map reduce, data flows, SQL, such as Apache Spark. To avoid the overheads of this task, we reused a predefined software stack, available as a Docker container. This involved three major steps: First we installed Docker, which is a computer program supporting operating-system-level virtualization, also known as containerization. A container consists of a standard unit of software and its dependencies so the application can run quickly and reliably from one computing environment to another. Second, we downloaded an image from Docker (https://hub.docker.com/r/jupyter/all-spark-notebook/), here we selected the ‘jupyter/all-spark-notebook’. This notebook includes Python, R, and Scala support for Apache Spark. By doing so, we ensured a flexibility to employ different scripting languages, and not to limit ourselves to one scripting language. Third, we developed and executed custom scripts (in Python and C++) via Jupyter Notebook (http://jupyter.org/) and for visualization we used IBM PixieDust. An overview of the software used is given in Figure 1.

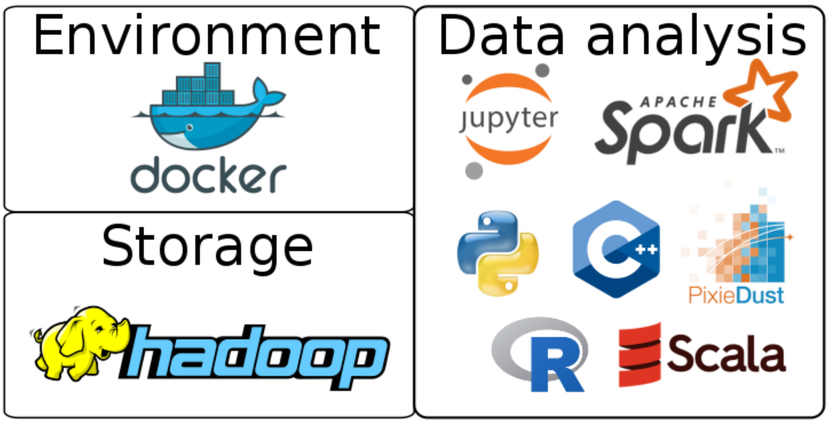

Figure 1. Overview of the software used in our data lake

Case study

Within the Breed4Food-Locomotion project, data from several sensors were collected during an animal experiment investigating the gait score of turkeys. These sensors included camera depth images (Intel Realsense D415), one force plate (Kistler), and three Inertial Measurement Units (Xsense MTw Awinda). These three IMUs were placed at the bird's legs (one each), and one IMU was placed on the bird's torso. Sensor-borne data were ingested to the data lake in raw format and stored using a Hadoop Distributed File System (HDFS). Sensor data of two selected sensors were binary data, i.e. force plate data (.tdms) and accelerometers (.mtb). In contrast, the raw output of the 3D-cameras was a bag file, this is a file format in Robot Operating System (ROS) for storing a sequence of records. Here we used the Melodic Meriona version of ROS to extract the data we needed for further processing.

ETL (Pre-processing and storage)

We generated customized scripts to convert the raw formats of each sensor to prepare these files for downstream analyses, see also Figure 2. For the force plate data, the technical data management solution (.tdms) files were converted to comma separated values (.csv) files via the cross-platform Python package npTDMS. The npTDMS package is created to read and write TDMS files as produced by LabVIEW. For the extraction of the accelerometer files (.mtb files), a C++ script was used to automatically extract the information and convert this to a .csv file. Lastly, the information from the 3D-cameras (.bag files) were extracted by using a custom-made Python script that links to the ROS.

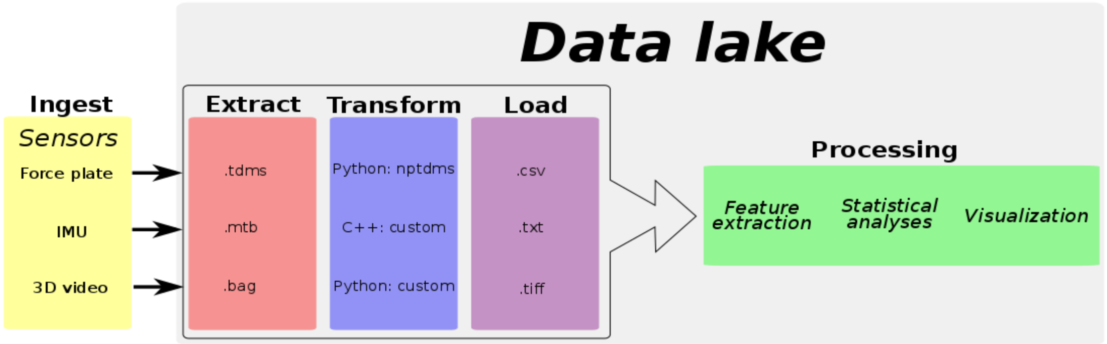

Figure 2. Flow diagram of the 'Extract, Transform, and Load' procedure of the data lake

Results and Discussion

Data lake

In animal sciences a plethora of data are generated and is expected to continue in the (near) future. Important aspects to take into account include how to ingest (the process of moving data into a distributed file system), store, process, and access these data. In the current study, we explored a data lake in which raw sensor data were collected and ingested, and subsequently transformed and visualized to investigate the benefits of a data lake approach, and to identify possible caveats of this approach compared to the more traditionally used structured and relational data bases.

Building and managing

Before starting to work with a data lake, some prerequisites are needed, like virtualization on your computer should be allowed. The advantage of building a data lake is that it can be exploited, for example when an (animal) experiment encompasses large volumes of data and it is necessary to scale-up the data ingestion. To this end, we built a customized data lake, designed to ingest the three types of sensor data, i.e. force plate, accelerometers (IMU), and 3D-video. The final deployable container image, an executable package including all necessary files, of the data lake was generated after multiple iterations. These iterations, although a labour intensive job, are of great importance because for scaling up this image needs to be deployed to many machines with no margin of error. The final image contained all necessary software and applications, as well as their dependencies, which was needed for transforming the animal experimental data that subsequently could be loaded in different programming languages, like Python and R. We could ingest and perform the ETL procedure for all animals at once and pre-process these data for further applications, analyses, and visualization. For example, all data was stored in its original format, allowing for extraction of alternative features in the future. Processing after a series of commands has been optimized and parallelized by the Apache Spark engine in the background. Another example is the possibility to link the different sensors, i.e. feature extraction of interest, in an analysis pipeline and subsequently perform statistical methods, e.g. linear regression.

Another important aspect associated to deploying a data lake we encountered, was the possible skills gap. For building and managing a container image, knowledge is required about command line code (such as Linux and/or Bash) and other programming languages (including Julia, Perl, and Python). This skills gap in people has already been identified (Gesing, Connor et al. 2015, Connor, Jacobson et al. 2016, Gibert, Horsburgh et al. 2018) and finding and maintaining people with the proper skill set is often more difficult in organisations compared to ‘processes’ and ‘technology’. In practice, this means that there is a shortage of persons with the necessary skills, this is also observed within the animal sciences domain, where to our knowledge not much effort is (yet) put into data lakes, and big data technologies in general. Thus, to adopt such approaches, investments need to be made, especially in acquiring the skills. Simultaneously, a start needs to be made by translating and adopting today’s challenges in the data lake approach. For example, by initiation of embedding this data lake approach into education of (MSc and) PhD students, and especially into animal experimentation. For the latter, it is expected that in the near future data from animal experimentation will become larger in volume and more complex. Compared to data warehouses, data lakes have a flexible configuration and high agility, as well as up to 10 to 100 times less expensive to deploy compared to conventional data warehousing (Stein and Morrison 2014, Khine and Wang 2018). Moreover, for precision livestock farming where it is expected that sensor data will become real-time in the (near) future, ingestion and storage of large volumes of data, can be performed immediately by a data lake approach. In our case study, the biggest challenges were the data transformations, due to the encrypted raw file formats from the different sensors. Nevertheless, we managed to overcome this by branching out to domain experts.

Metadata

Metadata is exceptionally important for managing your data lake. This encompasses both the earlier mentioned versioning of software and packages, as well as the description of the different data types, how these data were measured and under which circumstances. Generating such metadata is essential, as without effective metadata, data that streams into a data lake may never be seen again. Another important aspect getting more attention lately, is the Findable, Accessible, Interoperable and Reusable (FAIR) Guiding Principles (Wilkinson, Dumontier et al. 2016). For our data lake, we primarily focused on the accessibility of the data, using open source scripts and customize them to our data lake. By generating Jupyter Notebooks with extensive descriptions of the various commands it will be possible for other persons to reuse the data and scripts. However, more importantly, the various scripts could be read and executed by linking to other computers. In the current case study, we have generated a 'closed', IP-protected, FAIR data point, because we collaborate with an industry partner in the breeding sector.

Conclusions

The main lesson learned was that with a data lake approach it is possible to capture and maintain the entire universe of data from, e.g. an animal experiment in one virtual location (i.e. a container image). In addition, there is no data loss and the data are stored in raw format. This opens the possibilities to revisit the data and perform alternative pre-processing fitting novel or different hypotheses to the original stated hypothesis. Lastly, the whole procedure to extract, transform, and load (ETL-procedure) is scalable and could therefore reduce computing time, which is desired for dynamic analyses pipelines.

Acknowledgements

This study was financially supported by the Dutch Ministry of Economic Affairs (TKI Agri & Food project 16022) and the Breed4Food partners Cobb Europe, CRV, Hendrix Genetics and Topigs Norsvin. In addition, we want to thank Jeremie Vandenplas and Koen van Boheemen for their input in the scripts to transform the raw sensor data into text or tiff files. We are also grateful to the Breed4Food consortium for sharing the data and Aniek Bouwman for the fruitful discussions regarding the locomotion data.

References

Connor, C., A. Jacobson, A. Bonnie and G. Grider (2016). An Innovative Approach to Bridge a Skill Gap and Grow a Workforce Pipeline: *The Computer System, Cluster, and Networking Summer Institute. {USENIX} Journal of Election Technology and Systems* ({JETS}) 2(1).

Gesing, S., T. R. Connor and I. J. Taylor (2015). Genomics and Biological Big Data: Facing Current and Future Challenges around Data and Software Sharing and Reproducibility. CoRR abs/1511.02689.

Gibert, K., J. S. Horsburgh, I. N. Athanasiadis and G. Holmes (2018). Environmental Data Science. *Environmental Modelling & Software* 106: 4-12.

Khine, P. P. and Z. S. Wang (2018). Data lake: a new ideology in big data era. *ITM Web of Conferences*, EDP Sciences.

Stein, B. and A. Morrison (2014). The enterprise data lake: Better integration and deeper analytics. PwC Technology Forecast: Rethinking integration 1(1-9): 18.

Wilkinson, M. D., M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J. W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. t Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S. A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao and B. Mons (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data volume 3,* Article number: 160018.