My Signature Work (SW) was deeply influenced by three important courses: COMPSCI 301 (Algorithms and Databases), STATS 302 (Principles of Machine Learning), and COMPSCI 304 (Speech Recognition). These courses laid down the basic building blocks for my work, teaching me the crucial skills and concepts needed to tackle big challenges with smart solutions. The experiential learning (EL) part of my education at DKU was a mix of these key courses and hands-on lab work. This approach helped me connect what I learned in class to the real world, turning theories and ideas into tools for solving actual problems. The courses gave me a strong start in programming, database, machine learning, and dealing with time series data, each adding something to my SW. Learning about algorithms and databases gave me the tools to organize and find data efficiently. Machine learning opened up ways to understand and predict patterns in data when planning. Speech recognition brought a fresh way to look at how to process and make sense of live data, pushing me to think of new ways to approach problems in my SW. Lab works before and during the SW also contributed to the development of my skills for the SW and beyond.

Having taken the course COMPSCI 301 under gave me a strong foundation in Python programming and some understanding of database management, crucial skills that I later applied in my signature work. This course covered a wide array of essential topics, including sorting algorithms, dynamic programming, and an in-depth look at database systems, such as SQL, transaction management, and data recovery techniques, improving my computer science skills in general.

In the process of developing a system for monitoring gym equipment using MetaMotion sensors, my proficiency in Python was a great help. The programming skills honed during the course enabled me to effectively write and debug the software necessary for collecting and processing sensor data while official documentations are lacking. This task required not just programming knowledge but also experience as I have to figure out how to make the system work and debug it based on official examples and the result of my test runs.

Moreover, the database knowledge I acquired played a significant role in the architecture of the system. In the first version of the system, I was basically building a database system of my own through clever naming of the files on an FTP server. In the second and third version of the system I was using InfluxDB to store the data. While very different to SQL, basic concepts from database was still very useful to development based on InfluxDB.

In STATS 302, I learned about the basics of all kinds of machine learning algorithms and how to properly visualize data with Python. This course showed me how to take

real-world data and use it to make predictions, which greatly affect how did I plan my signature work initially. During the class, we went through the steps of figuring out a machine learning problem, picking the right tools to solve it, and then making it happen in Python. I got better at choosing the right algorithms for the right problem and making sure my programs could understand general patterns in data, not just memorize it by avoiding overfitting. Plus, the Kaggle project we did was a great way to test out what we learned in a real competition.

In the process of visualizing the acceleration data for my SW, my experience with STATS302 proved to be valuable. I used matplotlib to create clear, informative graphs that showed how the acceleration of gym equipment changed over time. This was crucial for spotting the trend when the equipment is in use. Moreover, when I worked with moving standard deviation data, matplotlib helped me illustrate how the result of this operation change based on window size and its behavior when there isn't enough data for one window. The experience in STATS 302 helps me make the graphs interactable and clean looking.

The machine learning skills I learnt from the class was mainly used during the conception phase of the SW. Back then we are thinking of taking advantage of the individual periods files formed from the division and employ semi-supervised learning based on labels from PHYSEDU classes. Though later down the line it would seem that I was seeing everything as nails after learning how to use the hammer that is machine learning. Later down the line it would seem that machine learning at least in the way I envisioned isn't the best way to handle the data.

COMPSCI 304 focused on the basics of speech recognition algorithms. It teaches me the skills to handle, compare and interpret audio data effectively. This course covered a broad spectrum steps and techniques used in speech recognition, including data capture, feature extraction, and advanced modeling methods like hidden Markov modeling and Gaussian mixture modeling. Through these topics, I learned how to manage and analyze sequential data that changes over time, which is a crucial aspect of understanding speech patterns and, by extension, any sequential and time series data. Some similar algorithms and techniques were also learnt in Duke COMPSCI 260 Introduction to Computational Genomics, but back then the versions we learnt weren't enough for making the connection to time series.

The hands-on approach of COMPSCI 304 was very helpful for my understanding. Through a series of progressively complex projects, I had the opportunity to build a classical speech recognition system from the ground up using Python. This experience taught me not just about speech recognition but also about managing time series data

in general—how to capture it, process it, and extract meaningful information from it. It also taught me how to handle programs with multiple threads.

Knowledge about multi-threading and processing techniques was an aspect of COMPSCI 304 that really contributed to the architecture of my signature work on monitoring gym equipment using MetaMotion sensors, especially in version 3. The ability to handle multiple threads is a necessity given the real-time nature of the sensor data being captured and handled in versions. This knowledge guided me to try to design a system that could simultaneously process incoming data streams from multiple sensors without bottlenecking, though the eventual effect might not be up to my vision due to incompatibilities between different multi-threading systems. Despite this, the insights gained from COMPSCI 304, complemented by the practical coding experience, were instrumental in analyzing the way in which the system goes wrong.

Knowledge about time series data from COMPSCI 304 is useful in two ways. One of the key insights came from the application of end pointing algorithms, commonly used in speech recognition to distinguish between periods of speech and silence within an audio stream. By adapting this algorithm to analyze the moving standard deviation of acceleration data from gym equipment, I was able to accurately identify when the equipment was in use. This methodology provided quick algorithm that can ,be applied in real-time on the flight. Moreover, the principles of Nyquist's theorem covered in COMPSCI 304 played a role in deciding the system's data acquisition strategy. Understanding that to accurately capture a signal without aliasing, the sampling frequency must be at least twice the highest frequency present in the signal, I decided that the maximum sampling frequency under streaming mode should be good enough for our application.

Other than the thematic courses, other EL comprise mainly of other related lab projects related to this one. Namely the particle tracking radar system for granular drag, the previous SRS smart gym project and the latter attempt to incorporate indoor AQI into the monitoring system.

From the particle tracking radar system for granular drag in zero-g, I have learnt how to program in Verilog. Also, the MetaMotion Sensors were introduced as a part of the alternative methods for studying that subject. For the previous SRS smart gym project, I was introduced to most parts of the system except for InfluxDB. Though I was able to achieve something similar to version 1, no data collection was done due to travel schedules and teamwork issues. It taught me how not to approach teamwork. The final project utilizes the adaptability of InfluxDB and showed me how the system could go in the future.

In conclusion, this project served as an introduction into the realms of Internet of Things (IoT) and software development, equipping me with practical know-how and insights that are crucial in these fields. By designing and implementing a system for monitoring gym equipment using MetaMotion sensors, I went through sensor-based data collection, real-time data processing, and network communication, which are foundational aspects of IoT solutions. The experience with programming in Python without good documentation, optimizing data flow, and managing a time-series database like InfluxDB challenged my software development skills, particularly in dealing with backend infrastructure and data analysis algorithms. This project improved my problem-solving mindset alongside with my technical knowhow.