Cyclist Weight Inference using IMU Sensors on Bicycles

CRISTIAN ANGHELUȘ, University of Twente, The Netherlands

Inertial Measurement Unit (IMU) sensors are widely used in various movable applications, including sports science, healthcare, and navigation. Recently, IMUs have also been used on bicycles to generate insights about road quality, fall detection, maneuver prediction, etc. IMUs are usually not known to contain sensitive data, but with the emergence of more advanced computer intelligence and machine learning techniques, we cannot be sure that certain sensitive insights, such as the weight of the cyclist, could not be inferred from the sensor data. This research investigated to what extent the cyclist's weight can be determined using only the IMU data. This study included collecting the IMU data from a cyclist with different added weights in a controlled experiment. Next, it analyzed and preprocessed the obtained data, followed by the development of models, using machine learning techniques, which can classify the weights of cyclists based on the IMU data. Finally, it evaluated the models and explored the feasibility of classifying the weights into an increasing number of classes. It was found that we can feasibly classify the weights up to 4 classes before the accuracy drops too low. We were able to achieve models able to classify the weights with up to 84% accuracy. The research is expected to contribute to advancing the understanding of privacy considerations when using sensor data as well as the possibilities of modern technologies to infer sensitive information from seemingly reliable and safe sensor data.

Additional Key Words and Phrases: IMU, bicycle, weight classification, machine learning, privacy, accelerometer, gyroscope, sensitive data, data inference.

1 INTRODUCTION

Bicycles are a popular mode of transport and a cornerstone of urban mobility, due to their convenience, sustainability, and health benefits. In countries like the Netherlands, cycling accounts for more than one-quarter (28%) of all trips [4]. With the advancement of modern technologies, new types of bicycles such as electric bicycles (e-bikes) and speed pedelecs are gaining more popularity. In this context, integrating various sensors into bicycles presents opportunities to generate and collect a lot of useful data that could be harnessed in multiple important applications. For instance, consider rental bike companies that could use the data to monitor their bicycles more extensively and improve their service, or governments that could leverage the data for assessing the road quality or improving the cyclists' safety in traffic. However, alongside the convenience and efficiency offered by these innovations, there are multiple concerns regarding the potential leakage of sensitive information from embedded sensors, particularly motion sensors [6, 8].

There have already been efforts to apply these technologies to the domain of cycling. In this sense, the increasingly advanced IMUs are promising for bicycles due to their low cost, simplicity, compactness, and low processing power [1]. IMUs are already used in various areas, such as manufacturing, healthcare, robotics, navigation, and sports learning [1]. Moreover, it's already been extensively used in various bicycle applications [3, 5, 10, 12, 13, 15]. However, there has been little research regarding the leakage of sensitive data from the IMU sensors, specifically in the context of bicycle applications. Most existing work focuses on other mobile devices and applications. While it is usually believed that IMUs do not contain sensitive data, we might underestimate the potential of advanced machine learning techniques, which might be used for sensitive information inference.

This is an important aspect that needs to be assessed, as it might have real implications in certain use cases. Consider the scenario where the rental bikes equipped with IMUs unintentionally leak sensitive information about cyclists, such as their weight, height, or location. Also, the organizations handling the data might compromise privacy and fail to implement adequate security measures as a result of their underestimation of the data value and potential.

This research will be focused on exploring the feasibility of inferring the cyclist's weight based only on the IMU sensor data. We will collect the data by conducting a controlled experiment, then we'll build models capable of classifying cyclists' weights using the obtained data, based on which we'll make the appropriate conclusions regarding the data inference possibilities.

This paper is organized as follows. Section 2 outlines our main research question and sub-questions. Section 3 reviews related research in the domain of sensitive information inference from sensor data and the usage of sensors on bicycles for various applications. Section 4 describes the methodology that we used to answer each of our research questions. Section 5 presents and thoroughly describes our obtained results. Finally, sections 6, 7, and 8 present the discussion, conclusion, and future work, respectively.

2 RESEARCH QUESTIONS

Our introduction and problem statement lead to the following formal research question:

To what extent can we determine the weight of a cyclist using only the IMU sensor data mounted on the bicycle?

This general question leads us to the following sub-questions:

(1) How can we collect data relevant to the cyclist's weight from the IMU sensors?
(2) How can we use the machine learning models to infer the weight of a cyclist using the IMU data?
(3) How do the ML models compare in terms of performance?

3 RELATED WORK

In this section, we review some existing research relevant to the inference of sensitive information from sensor data, as well as studies focusing on the usage of sensors on bicycles for various applications. In their paper, Huang et al. [8] assessed the risk of keystrokes and

TScIT 41, July 5, 2024, Enschede, The Netherlands

full-text input inference from the motion sensors (including accelerometers and gyroscopes) of Android devices. For inference, they developed a machine learning model, specifically a Deep Fully Conventional Neural Network (DFCNN) that achieved a high enough accuracy to eavesdrop on users' sensitive information, such as account numbers, passwords, and other exact values. In another paper, Naval et al. [11] used the motion sensors' data (accelerometer, gyroscope, orientation sensors) to infer 4-digit PINs from smartphones embedded with such sensors. In their approach, they also used machine learning models, such as MLP, Decision Tree, Random Forest, and Naïve Bayes. Also, Han et al. [7] used the accelerometers on smartphones to infer the location of the device owner within a 200-meter radius of the true location, while Van hamme et al. [14] studied the feasibility of predicting age and gender with reasonable accuracy on gait traces using the IMUs sensors. In their study, they also employed machine learning models for feature extraction as well as for the predictions. Additionally, Kröger et al. [9] presented an overview of even more possible inferences from accelerometer data, including user's activities, location, device inputs, moods/emotions, or personality traits.
These papers highlight the existing potential vulnerabilities of inferring sensitive data from motion sensors, further justifying the scope of our research. Additionally, they can serve as valuable reference points for our methodology, specifically for the data analysis and ML model selection and development process.

In the domain of bicycles, there has been substantial research regarding the usage of IMUs mounted on bicycles for various applications. For example, de Smit et al. [5] used IMUs to predict the turn maneuvers of cyclists to help prevent collisions, Zhang et al. [15] used the sensors to estimate the pose of the rider/bicycle to better study unstable physical human/robot interaction or human rider's gait, Mihaldinec et al. [10] used them to estimate the cycling cadence that could provide better personal health insights and assistance in training, Tabei et al. [12] tried to develop an accident detection system for cyclists using a variety of sensors to improve the cyclist's safety on the road, while Chang et al. [3] proposed a system to improve the cycling navigation using the IMUs from portable devices. These papers highlight the potential benefits of using the data from IMUs on bicycles for various applications. However, most of these papers assume that the data from the sensors is non-intrusive, which might not be entirely true, thus enhancing the importance of our research. Additionally, these papers can be used as reference points in our research, including the placement of sensors on the bicycle during data collection or the selection of appropriate ML models for our task.

Finally, it's worth noting that similar research to ours was already done by another student. In his research, Benitah [2] also explored the possibility of inferring the cyclist's weight using the IMU data, however, we claim that his methodology was partly flawed. Specifically, for his data collection, he used multiple (12) participants for the experiment. This introduced more variables in the experiment and resulted in an unevenly distributed dataset, which highly influenced the accuracy and correctness of the models. In our approach, we propose an entirely different methodology. First, we will use only one cyclist for the experiment, with a larger set of additional weights on him, so we'll simulate different weight classes with one cyclist. By considering a single cyclist, we ensure consistency in data collection, and a more evenly distributed dataset as we reduce the variability that is introduced by different individuals. In the end, we aim to obtain a more controlled environment for the experiment, which should translate to better outcomes. Secondly, during the development of the models, previous research used the user ID as a training feature, which is considered wrong since the models will learn more based on that ID, rather than the actual data from the IMU. In our approach, we'll avoid this mistake and investigate more suitable models as well, which could lead to more accurate results. Also, we use different IMU sensors, which are better synchronized, more accurate, and easier to use.

Fig. 1. ProMove-mini IMU node

In summary, the described papers emphasize the motivation of our research and its importance. In addition, these papers will serve as valuable guidelines and insights for our decisions throughout the research.

4 METHODOLOGY

This section will detail the methodology that we used to answer each of our proposed research questions. The process includes data collection through a controlled experiment, data analysis and preprocessing, the application of machine learning models, and performance evaluation. This section is organized into subsections that present the approach used to answer each research (sub-)question.

4.1 On answering the 1st sub-question

To answer our first sub-question: How can we collect data relevant to the cyclist's weight from the IMU sensors, we decided to conduct a controlled experiment, in which we acquired the necessary data from the IMUs. The experiment consists of configuring and strategically attaching several sensors on a simple bicycle, collecting the data through a cyclist making multiple trips on the bike with different added weights and at different speeds, and recording and saving the data captured by the sensors. Next, we describe each of these parts in more detail.

4.1.1 Sensor Placement and Configuration. The sensor device used for this project is the ProMove-mini wireless IMU, which is a wireless networked inertial and orientation sensor. See Figure 1 for the visual representation of one such sensor node. For the scope of our research, we have decided to use simultaneously 3 such sensors that

Fig. 2. Weight vest used to simulate different weights

were placed on different parts of the bicycle. Each of the sensors was configured to record only the accelerometer and gyroscope data along the 3 axes (x, y, z). The sensors were set at 200Hz, which means that the sensors recorded 200 samples of data per second. The placement of the sensors was chosen as follows:

(1) Head Tube: This location was chosen because it captures the data from the front part of the bicycle, specifically it should capture the steering movements of the wheel, which we believe can be highly influenced by the cyclist's weight.
(2) Under the Seat: This placement was selected as it should capture the movements and vibrations transmitted through the seat post and the impacts experienced by the saddle, which should carry a significant portion of the cyclist's weight, as the cyclist sits directly on it, providing useful data related to the cyclist's weight.
(3) Rear frame: This position was chosen because it captures the data from the rear part of the bicycle, which also bears a significant portion of the rider's weight, especially during acceleration.

See Appendix A for pictures of the bicycle setup, including the placement of the sensors.
We believe that, by using 3 sensors that record data from different parts of the bicycle, we can get a more complete set of data, which allows us to analyze the inference of the cyclist's weight from different parts of the bicycle, and even see which one is more indicative of the weight.

4.1.2 Data Collection. As mentioned in Related Work section, our methodology focuses on collecting the data from one cyclist, which ensures consistency in data collection, a (relatively) evenly distributed dataset, and a more controlled environment overall. Our experiment requires several additional considerations, which are stated in Table 1, along with their motivation and our solution.

Overall, we performed the data collection for 16 different weights at 3 different speeds each, resulting in a total dataset of 48 different rides. Since the data collection process lasted multiple days, it resulted in slightly different weights between the speed categories, therefore we treat each weight from a ride as a separate data point.

Fig. 3. Selected road type for data collection

4.1.3 Recording and storing the data. For recording and storing the data, we used the internal memory of the sensor nodes. Generally, the sensor nodes are provided with a software tool from the manufacturer that allows for plot visualizations of the data, as well as synchronization between multiple sensor nodes and automatic recording of the data. However, this tool only works if the sensors are within the range of the gateway connected to the laptop via USB, but when we performed our data collection, the cyclist would very soon get out of range, therefore, we used the manual recording for each sensor.

Before each ride, the cyclist would manually turn on the recording of each sensor node by double-clicking the button on each sensor. After the ride, the cyclist would again double-click the button to stop the recording. The recordings of the sensors from each ride are thus saved in the internal memory of the nodes in a separate file. Then, after the rides, we would export the files from the internal memory of the nodes and convert them to .csv files by using the software tool.

As a result, we obtained 48 different CSV files for each speed (16 weights from 3 sensors), each containing the timestamp, node-id, ax, ay, az, gx, gy, and gz columns, that we could use further in our analysis and development of the models.

4.1.4 Experiment Setup. In this section, we will discuss the exact setup of the experiment. Before the ride, we first made sure that all the sensors were charged and firmly attached to the bike. Then, the cyclist would measure his weight right before the ride using a simple scale, with all the necessary added weights on him. This weight would then be recorded and stored separately. Then the cyclist would turn on manually the recording of each sensor on the bike, as well as the speedometer. The cyclist would then start the ride at the specified speed for exactly 5 minutes on the designated segment of the road, keeping track of the duration and speed of the ride from the speedometer. After 5 minutes of riding, the cyclist would stop immediately and manually turn off the recording of each sensor node. Then, he would go to the starting position and repeat the process with a different weight and possibly at a different speed.

4.2 On answering the 2nd sub-question

To answer our second sub-question: How can we use the machine learning models to infer the weight of a cyclist using the IMU data, we propose 2 approaches: feature-based approach and deep-learning

Table 1. Experiment considerations

Consideration	Motivation	Solution
Weight Simulation	To collect data produced by different weights but use only one cyclist, to exclude other variables.	The cyclist wore a weight vest, similar to the one in Figure 2, allowing several weight blocks to be added in its pockets. We decided to collect data using 16 different weights, including the base weight of the cyclist (without the weight vest) and 15 different added weights, resulting in a range of weights between 84.5 - 105.7 kgs.
Speed Variation	The speed of the cyclist can greatly influence the collected data, therefore we need to consider it as another variable.	We considered 3 speed classes: slow, medium, and fast. We didn't set an exact speed for each category, as maintaining a constant speed turned out to be challenging. Therefore, we set an approximate range for each speed category, which the cyclist would follow during all the trips. Also, we recorded the average speeds after each ride. These ranges can be observed in Table 2. To monitor and adjust the speed during the rides, the bicycle was equipped with a speedometer on the handlebar, visible to the cyclist, as seen in Appendix A.
Road Selection	To ensure consistency and to collect more complete and accurate data.	All the rides were performed on the same road segment, covered in circles if needed. To collect more accurate and complete results from the sensors, especially from the gyroscope, we also decided to select a road type that contains multiple bumps and turns. As a result, we decided to select a circle portion of the road type, as presented in Figure 3.
Duration of the Ride	To ensure that we have enough collected continuous data from each ride.	We set the duration of each ride to exactly 5 minutes. The duration was monitored by the cyclist using the speedometer. After 5 minutes of cycling, the cyclist would stop, regardless of the covered distance.
Cyclist's Position and Posture	To ensure consistency throughout the rides with different weights.	The cyclist was instructed to straighten their back and bend only as necessary to grab the handlebars and try to keep a constant position throughout the entire ride.

Table 2. Speed ranges and average speed

Speeds	Range	Average speed
Slow speed	7.5 - 9.5 km/h	8- 8.3 km/h
Medium speed	12 - 13.5 km/h	13 - 13.5 km/h
Fast speed	16 - 20 km/h	17 - 17.5 km/h

approach. Both of these approaches would also require some data preparation and preprocessing. These are further explained in the following sections.

4.2.1 Data preparation. To use our collected data for the development of our models, we first need to preprocess and transform the data to be suitable for our task. We decided that, for both approaches, we would build the models for each speed separately and then compare the results. For our classification task, we decided to initially classify the weights into 3 classes: light, medium, and heavy. For this reason, we nearly evenly split the 16 weights from each speed into the 3 classes. Then, we further split the data into training and testing sets. We used 2 weights from each weight class for testing and the other weights were used for training the models. We also transformed the raw time-series sensor data using the 'windowing' technique, as standard classification algorithms cannot be directly applied to raw time-series data. Therefore, we split the data into windows of 2 seconds, each containing 400 samples (200 samples/s). For labeling the samples, we used the most-occurring weight class from the window. Additionally, we considered overlapping windows with 50% overlap, instead of discrete windows. This ensures that every subsequent window in the transformed dataset contains some information from the previous window as well, preserving more correlations and continuity in the data, which is important for time-series analysis. This approach also helped us artificially increase the dataset for training and testing the models.

4.2.2 Feature-based approach. In this approach, we aimed to extract several features from the raw sensor data and then build some standard classification models based on the identified features. We explored both some simple statistical features and features in the frequency domain by performing a Fast-Fourier transform on the data (FFT).
For this approach, we selected 4 classification models: KNN, Decision Tree, Support Vector Machine (SVM), and Logistic Regression due to their complementary strengths and reviewed literature. KNN was used for its simplicity and ability to classify based on the similarity of data, rather than hyperparameters. Decision Trees can handle well non-linear relationships and are relatively robust against data

noise. SVMs are known for their ability to handle high-dimensional data and find optimal decision boundaries. Finally, Logistic Regression offers a straightforward, computationally efficient, probabilistic approach that is widely used for multi-class classification tasks.

4.2.3 Deep-learning approach. In the deep-learning approach, we wanted to see if we could leverage the raw sensor data directly, without manual feature engineering, allowing the models to automatically learn and extract relevant features.
To achieve this, we leveraged some simple deep-learning models, specifically a Convolutional Neural Network (CNN) model and a Multilayer Perceptron (MLP) model. CNN was used for its ability to capture spatial, temporal, and hierarchical structures from a sequence of IMU data and automatically learn relevant features, as well as for its simplicity and computational efficiency. MLP was selected for its simple architecture and capacity to model complicated, non-linear relationships between features and target classes.
Both these models can learn intricate patterns in the data and have the potential to achieve great results compared to the manually extracted features.

4.3 On answering the 3rd sub-question

To answer our third sub-question: How do the ML models perform and compare in terms of performance, we decided to build multiple models based on different data, like speed, sensor-id, and the type of collected data (accelerometer and gyroscope). Then, for each model, we made a classification report and confusion matrix. We assessed the performance based on the reported accuracy and f1 scores, but we also examined the precision and recall scores.

4.4 On answering the main research question

Finally, to answer our main research question: To what extent can we determine the weight of a cyclist using only the IMU sensor data mounted on the bicycle, we decided to extend our classification task to include multiple weight classes instead of just 3. This allowed us to evaluate the performance of our models when attempting to classify the weights into more precise classes.
To achieve this, we chose the models with the highest accuracy from our initial 3-class classification setup and then used them to classify the weights into a range of classes, from 2 to 8. As a result, we could assess the performance of these models and make conclusions about the extent of our classification possibilities.

5 RESULTS

In this section, we present the results that we obtained after following our described methodology. First, we'll describe our analysis of the obtained data from the experiment, as well as the necessary preprocess that was done. Then, we'll describe how we developed our feature-based and deep-learning models, and present the evaluation of their performance. Finally, we'll describe the results obtained from multiple classes classification.

5.1 Data Analysis & Preprocess

After conducting the data collection experiment, we examined our obtained dataset. As mentioned in the Methodology section, we obtained a dataset of 48 data points (16 weights at 3 speeds). Then, for each of them, we have data from 3 sensors simultaneously. The weight distribution can be seen in Figure 4.

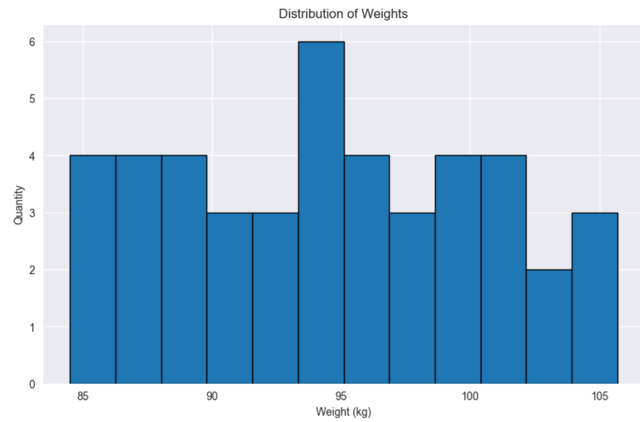

Fig. 4. Weight Distribution

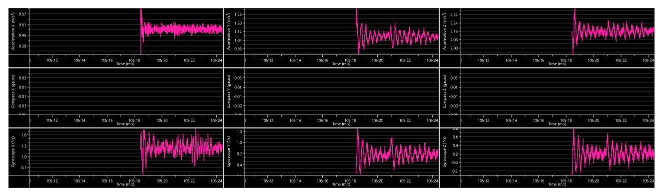

Fig. 5. Example of the real-time replay of a ride in Inertia Studio

Next, we analyzed all the obtained files in the Inertia Studio (the software app for the sensors), which allows for real-time replay of the ride. An example of how we can visualize the data is presented in Figure 5. While checking the files, we noticed that the actual ride starts a few seconds after the start of the recording. This is because the cyclist had to turn on manually each sensor, and then after adjusting himself on the bike, he would start the 5-minute ride. This resulted in different amounts of samples recorded from each ride. To ensure the consistency and even distribution of the data, we decided to keep exactly 5 minutes of data from each ride, by deleting the first samples from each file, which corresponded to the adjusting and calibrating time before the start of the actual ride. Apart from this, the manual inspection of the files did not reveal any visible patterns or observable conclusions.

5.2 Feature-based Models

After analyzing and preparing the data, we first built the feature-based models, as described in the Methodology. In total, we identified 94 features that provided relevant information for training the models. This included statistical features[1] such as the mean, median, min, and max, for each of the 3 axes. Then we explored most of these features again, but in the frequency domain, by applying FFT to the

[1]Article containing the features that we used: https://towardsdatascience.com/feature-engineering-on-time-series-data-transforming-signal-data-of-a-smartphone-accelerometer-for-72cbe34b8a60

Table 3. Accuracy scores for Medium Speed

Model/Sensor Placement	Head Tube	Under Seat	Rear Frame	All sensors
KNN	0.4	0.64	0.36	0.43
Decision Tree	0.41	0.75	0.38	0.5
SVM	0.53	0.82	0.42	0.49
Logistic Regression	0.54	0.84	0.42	0.49

Table 4. Accuracy scores for different speeds

Model/Speed	Slow speed	Medium speed	Fast speed
KNN	0.67	0.64	0.57
Decision Tree	0.74	0.75	0.71
SVM	0.75	0.82	0.77
Logistic Regression	0.76	0.84	0.78

data. For each speed, we also developed separate models using the data from each sensor node separately, as well as from all 3 sensors combined. For our features, we also experimented with using only the accelerometer data, only the gyroscope data, and a combination of both. However, in all the models, the gyroscope data resulted in a much lower accuracy compared to the accelerometer data, therefore we only used the accelerometer data for extracting our features.

Table 3 presents the accuracy scores of each model for medium speed and different sensors. As can be observed, the sensor placed under the seat produces far better results in terms of accuracy for each model. This was also the case for both the slow and fast speeds. Therefore, we present in Table 4 the accuracy scores for the different speeds using only the sensor under the seat. From these results, we can observe that the medium speed generally produced the highest accuracies. Also, the KNN performed the poorest for all speeds, with around 60% accuracy. The Logistic Regression and SVM models performed the best, being quite close to each other, while the Decision Tree also performed well, with around 70% accuracy, and was close to SVM and Logistic Regression, especially for low and high speeds.

5.3 Deep-learning Models

After implementing the feature-based models, we proceeded to develop the deep-learning models to explore the feasibility of classifying the weights using the raw data instead of manually extracting relevant features.

5.3.1 CNN Model. For the CNN model, we developed a simple 1-dimensional model. The model architecture summary is represented in Figure 6. As in the feature-based approach, we developed separate models for each speed and sensor. Also, we experimented with both the accelerometer and gyroscope data, however using only gyroscope data again resulted in much poorer accuracy, while using only the accelerometer data we achieved the best accuracy. Therefore, we present the results from using only the accelerometer data. Table 5 presents the accuracy scores of the model for different speeds and sensors. As with the feature-based approach, we can observe that we obtain the highest accuracy when using the data from the sensor under the seat. Also, the medium speed produced higher accuracies.

Table 5. CNN Accuracy Scores

Speed/Sensor Placement	Head Tube	Under Seat	Rear Frame
Slow speed	0.47	0.73	0.42
Medium speed	0.54	0.84	0.4
Fast speed	0.42	0.61	0.49

Fig. 6. CNN model

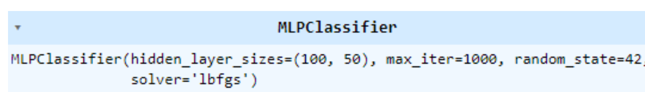

Fig. 7. MLP model

Table 6. MLP Accuracy Scores

Speed/Sensor Placement	Head Tube	Under Seat	Rear Frame
Slow speed	0.43	0.73	0.46
Medium speed	0.37	0.76	0.38
Fast speed	0.39	0.66	0.39

The highest obtained accuracy was 84%. We can also state that the model performed poorly at the fast speed, achieving the highest accuracy of 61%.

5.3.2 MLP Model. For the MLP model, we developed a simple MLP-Classifier using the scikit-learn[2] package for Python. We experimented with the model's parameters and decided on the final version represented in Figure 7. As with the CNN model, we used only the

[2]https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

Table 7. Accuracy scores for different number of classes

Model/Classes	2	3	4	5	6	7	8
SVM	0.80	0.82	0.70	0.60	0.5	0.35	0.38
Logistic Regression	0.81	0.84	0.70	0.63	0.48	0.35	0.37
CNN	0.86	0.84	0.64	0.41	0.41	0.31	0.30
MLP	0.91	0.73	0.61	0.45	0.43	0.34	0.33

accelerometer data for training the model. Table 6 shows the accuracy scores of the model for different speeds and sensors. Here we can observe that the under-seat sensor still achieves better accuracies, however, the difference between the speeds is smaller, with the fast speed achieving the worst accuracies, and the best accuracy still achieved by the medium speed, at 76%.

5.4 Multi-class Classification

After developing all of our models based on a 3-class weight classification, we proceeded to train the models to classify the weights into more classes. Based on the performance of the initial models, we can state that we achieved the best results using the data from the under-seat sensor at medium speed. Therefore, we used only this data to train our models on multiple classes.

Table 7 reflects our obtained results. For the feature-based models, we present the results only from the best-performing models: SVM and Logistic Regression. From the results, we see that the accuracy drops significantly as we increase the number of classes. After 4 classes, the models' accuracies are too low (<60%) to consider them effective, since such low accuracies are equivalent to simply guessing the weight class. We can also notice that the feature-based models (SVM and LR) are performing slightly better as we extend the number of classes, while the deep-learning models are still performing well for fewer classes (2 and 3), and then we observe a more significant drop in accuracy.

6 DISCUSSION

In this section, we will discuss and explain our findings, based on the results from the previous section.

Firstly, we can state that we found a relationship between the cyclist's weight and the IMU sensor data collected from their rides. Our results showed that the accelerometer data was more indicative of the weight rather than the gyroscope data, which performed much worse in terms of the accuracy of the models. Our findings indicate that the optimal sensor placement and speed significantly impact the performance of weight classification models. The sensor placed under the seat consistently outperformed other sensors. The medium speed slightly outperformed the slow and fast speeds most of the time.

Secondly, we can say that it is possible to classify the weights using both feature-based and deep-learning models. The models perform relatively similarly well for classifying the weights into 3 classes. The deep-learning models' performance also indicates that it is possible to automatically learn features from raw sensor data, rather than manually engineer them.

Thirdly, as we increase the number of classes for the classification task, the accuracy drops consistently and significantly, which might show the limitations of our classification task. Perhaps, if we want to determine more precisely the weight, a regression task of prediction might be more suitable.

7 CONCLUSION

This research aimed to determine the extent to which the weight of a cyclist can be inferred from IMU sensor data mounted on a bicycle. We addressed this through a series of sub-questions that provided valuable findings and helped answer the main research question.

Regarding the first subquestion, it can be concluded that we can collect data relevant to the cyclist's weight from the IMU sensors through a controlled experiment. Our results indicated that placing the sensor under the saddle provided consistently the highest accuracy of the classification models. Additionally, accelerometer data proved more useful than gyroscope data, providing better accuracy of the classification models. Also, the speed of the cyclist influenced the performance of the models. We found the medium speed to be overall the most effective for data collection. These variables indicate the most effective setup for data collection, capturing data that is most indicative of the cyclist's weight.

Regarding the second subquestion, we showed that we can infer the weight of the cyclist by manually extracting relevant features from raw sensor data and training traditional machine learning models (KNN, Decision Tree, SVM, and Logistic Regression) to classify the weights into multiple classes. We also explored deep-learning classification models (CNN and MLP) and showed that they are capable of capturing relevant features regarding the weight directly from the raw sensor data. Both approaches demonstrated good feasibility of weight inference, and are quite close in terms of accuracy of the classification task.

To answer the third research question, we evaluated all the developed models in terms of their accuracy scores, as well as by analyzing their precision, recall, and f1-scores. We found that, for a 3-class classification task, the CNN and the Logistic Regression models performed the best, achieving an accuracy of 84%.

Addressing the main research question, our study shows that it's feasible to classify the weights of a cyclist into more precise classes, however, the accuracy of the classification consistently diminishes as we increase the number of classes and the classification task becomes more precise. Our results revealed that we can feasibly classify the weights up to 4 classes. After that, the classification models become irrelevant, as their accuracy drops below 60%, which is effectively the same as guessing the weight class, rather than classifying it.
Nevertheless, these results already show great possibilities of inferring the cyclist's weight and should increase the concern over the protection of their privacy.

8 FUTURE WORK

Although our research shows promising results, there are still areas of improvement for future work.

Firstly, increasing the amount of collected data and the environment of the experiment to be even more controlled could always lead to more interesting and relevant findings.

Secondly, in our research, we mentioned that the gyroscope data did not provide relevant data, however, there is a possibility to consider the data only from a turn of the bicycle. During the turn, the gyroscope data should capture more relevant data as this is when it will record more rotations. Additional research into the possibilities of gyroscope data can greatly improve the performance of the models.

Finally, instead of performing a classification task, further research might consider a prediction task, such as regression, where the models should predict the exact weights, rather than classifying them. This can further improve the possibility and the extent of inferring the weights.

REFERENCES

[1] Norhafizan Ahmad, Raja Ariffin Raja Ghazilla, Nazirah M. Khairi, and Vijayabaskar Kasi. 2013. Reviews on Various Inertial Measurement Unit (IMU) Sensor Applications. *International Journal of Signal Processing Systems* (2013), 256–262. https://doi.org/10.12720/ijsps.1.2.256-262

[2] Remy Benitah. 2022. Cyclist Weight Inference from Bicycle-Mounted Sensor Data. *Proceedings of Twente Student Conference on IT (TScIT 38)* 1 (2022). https://doi.org/10.1145/nnnnnnn.nnnnnnn

[3] Hsiu-Wen Chang, Jacques Georgy, and Naser El-Sheimy. 2012. Improved Cycling Navigation Using Inertial Sensors Measurements From Portable Devices With Arbitrary Orientation. *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT* 64 (2012). Issue 7. https://doi.org/10.1109/TIM.2014.2381356

[4] M.C. De Haas and L. Kolkowski. 2023. Cycling Facts 2023 Brochure.

[5] Gijs de Smit, Deepak Yeleshetty, Paul J.M. Havinga, and Yanqiu Huang. 2024. Predicting Turn Maneuvers of Cyclists Using Bicycle-Mounted IMU with CNN-LSTM. *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 587–592. https://doi.org/10.1109/PerComWorkshops59983.2024.10502681

[6] Paula Delgado-Santos, Giuseppe Stragapede, Ruben Tolosana, Richard Guest, Farzin Deravi, Ruben Vera-Rodriguez, Ruben Vera, ; G Stragapede, R Tolosana, and R Vera-Rodriguez. 2022. A Survey of Privacy Vulnerabilities of Mobile Device Sensors. *Comput. Surveys* 54 (2022). Issue 11s. https://doi.org/10.1145/3510579

[7] Jun Han, Emmanuel Owusu, Le T. Nguyen, Adrian Perrig, and Joy Zhang. 2012. ACComplice: Location inference using accelerometers on smartphones. In *2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012)*. 1–9. https://doi.org/10.1109/COMSNETS.2012.6151305

[8] Yan Huang, Student Member, Xin Guan, Hongyang Chen, Senior Member, Yi Liang, Shanshan Yuan, and Tomoaki Ohtsuki. 2020. Risk Assessment of Private Information Inference for Motion Sensor Embedded IoT Devices. *IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE* 4 (2020). Issue 3. https://doi.org/10.1109/TETCI.2019.2902866

[9] Jacob Leon Kröger, Philip Raschke, and Towhidur Rahman Bhuiyan. 2019. Privacy implications of accelerometer data: a review of possible inferences. In *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy* (Kuala Lumpur, Malaysia) *(ICCSP '19)*. Association for Computing Machinery, New York, NY, USA, 81–87. https://doi.org/10.1145/3309074.3309076

[10] Hrvoje Mihaldinec, Luka Klaić, and Hrvoje Džapo. 2023. Estimation of the Cycling Cadence Using Low-Cost Inertial Sensors Mounted on a Bicycle. *2023 6th Conference on Cloud and Internet of Things (CIoT)*, 24–28. https://doi.org/10.1109/CIoT57267.2023.10084907

[11] Smita Naval, Akanksha Pandey, Shivam Gupta, Gaurav Singal, Vignesh Vinoba, and Neeraj Kumar. 2022. PIN Inference Attack: A Threat to Mobile Security and Smartphone-Controlled Robots. *IEEE SENSORS JOURNAL* 22 (2022). Issue 18. https://doi.org/10.1109/JSEN.2021.3080587

[12] Fatemehsadat Tabei, Behnam Askarian, and Jo Woon Chong. 2021. Accident Detection System for Bicycle Riders. *IEEE SENSORS JOURNAL* 21 (2021). Issue 2. https://doi.org/10.1109/JSEN.2020.3021652

[13] Yuri Usami, Kazuaki Ishikawa, Toshinori Takayama, Masao Yanagisawa, and Nozomu Togawa. 2018. Bicycle Behavior Recognition using Sensors Equipped with Smartphone. In *2018 IEEE 8th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*. 1–6. https://doi.org/10.1109/ICCE-Berlin.2018.8576254

[14] Tim Van hamme, Giuseppe Garofalo, Enrique Argones Rúa, Davy Preuveneers, and Wouter Joosen. 2019. A Systematic Comparison of Age and Gender Prediction on IMU Sensor-Based Gait Traces. *Sensors* 19, 13 (2019). https://doi.org/10.3390/s19132945

[15] Yizhai Zhang, Fei Liu, Mitja Trkov, and Jingang Yi. 2012. Rider/bicycle pose estimation with IMU/seat force measurements. In *2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. 604–609. https://doi.org/10.1109/AIM.2012.6265934

A BICYCLE SETUP

Below are images reflecting the bicycle setup that was used during the data collection experiment.

Fig. 8. Head Tube sensor placement

Fig. 9. Speedometer Placement

Fig. 10. Under-seat sensor placement

Fig. 11. Rear frame sensor placement