**Monte Carlo MUC**

**CS 4605 Monte Carlo MUC: Report**

Alberto Molina Felipe

Georgia Institute of Technology  
Atlanta, GA

afelipe3@gatech.edu

Joshua Rosenthal

Georgia Institute of Technology  
Atlanta, GA

jrosenthal8@gatech.edu

Jason Yang

Georgia Institute of Technology  
Atlanta, GA

jasonyang@gatech.edu

**ABSTRACT**

In the 1960s the first wearable computer was constructed by Edward Thorp and Claude Shannon to predict an outcome in roulette and grant an advantage to the user [6]. Sixty years later, with the advancements in computing technology, we attempted to remake and improve Thorp and Shannon’s system. Although the algorithm only found accuracy above random prediction in a constrained condition, the hardware implementation demonstrated an ease of development, comfort, and adaptability that comes from modern technology.

# INTRODUCTION

In an American roulette wheel, any single-slot bet results in an expected 5.26 percent advantage to the casino. This house edge guarantees that the average gambler will lose money if they gamble on a roulette outcome. With such a small edge, any way to increase a gambler’s odds of winning to over 50% could theoretically allow for massive success since they are now favored over the casino. Roulette is seen as a game of pure chance, but this paper presents the development of a device to predict the outcome of a roulette spin and beat the odds, all while being disguised from use.

The device takes inspiration from similar devices in the past, mainly the first wearable computer developed by Thorp and Shannon [6]. However, their original device was made almost 50 years ago, so it is worth evaluating the problem again with modern technological developments to improve on both the hardware and software aspects of the device. While this paper failed to improve upon their system’s prediction accuracy, it greatly improves upon the physical system for controlling such a device by making it more covert, more comfortable, and easier to use.

# BACKGROUND

This paper’s device is primarily based on Thorp and Shannon’s wearable device, where the pair built the first wearable computer in 1961 with the purpose of winning at roulette [6]. Their system worked by timing the spin of both a roulette table’s rotor (the piece with the ball slots) and the rotation of the ball around the table for each betting round [5]. These values were calculated using a microcontroller in the shoe with data supplied by the wearer’s toe taps; the system then calculated the ideal octant to bet using the inputted data and it was sent to an earpiece, allowing the duo to create “an expected gain of +44%” when betting in a casino [6].

While Thorp and Shannon were the first to create a roulette-predicting device, others have developed similar technology inspired by their work. In the 1980s, a group called the Eudaemons built a system for predicting roulette outcomes that worked on similar principles of calculating ball and rotor speeds [1, 2]. The key difference was that communication was done via vibrations in the player’s shoe [2], which is a far better disguise than Thorp and Shannon’s earpiece. Notably, the final design also hid the input computer in a hollowed-out shoe, although early attempts disguised the device in the user’s armpit [1]. According to recounts from Farmer, the person who designed the shoe, the system had severe and persistent hardware problems but still achieved a 20% edge over the house [1].

Modern developments in technology and society at large make it the perfect time to create a new version of the roulette calculator. First of all, the constant miniaturization of technology allows for these kinds of devices to be smaller and therefore even more covert. While previous groups needed to hollow out shoes to hide their highly specialized computers [2, 5], tiny microcontrollers are both easy to come by and work with for an average consumer. In addition, the growing prevalence of computers such as mobile phones and smart watches makes it even easier to disguise the device. It is even possible with Bluetooth to use those devices as inputs and/or outputs for a roulette prediction system, although this paper does not explore that avenue.

# DESIGN PROCESS AND DECISIONS

Thorp and Shannon took eight months to design and construct their device while we only had two months to complete the same work [6]. As a result of the rapid development process, many design decisions prioritized creating a minimum viable product and initial framework upon which future development could be easily done. Development of hardware and software were concurrently developed with clear guidelines and interfaces to ensure seamless interlayer communication and facilitate adaptability to changes in protocols.

## User Interface

The user interface experimentation preceded any construction of the hardware prototypes as this directed much of the structure and design of the physical device. The configurations were quickly tested with the group members using simple setups such as vibration motors held against the foot to allot more time for iteration of the mechanical design.

### Input

For user input, the device relies on a single button. This is pressed five times in a loop, three times to record the speed and friction of the ball and two times to record the speed of the wheel.

The use of the input did not deviate from the original plan; however, the input device changed to a button from an accelerometer. The vision during the conception was to have the user’s foot have subtle motions in four directions: up, down, left, and right. These motions would be silent, inconspicuous, and intuitive. The problems with using an accelerometer were pinpointing the exact moment the input would be taken, reliability of input, and feedback. An accelerometer’s motion requires filtering and thresholding to acquire an activation time point, which may introduce latency and inconsistency across different users as opposed to a simple button press. Additionally, it introduces a need for feedback through other components as well as a lack of trust which a button accomplishes through its simplicity and tactile click.

### Output

Vibration motors provide the user interface for the device’s output. After experimentation to ensure the user can easily identify the state the device is in as well as interpret the encoded output of the sector, a few major decisions were made regarding the correct way to utilize the motor hardware and the signal formats.

Although the user could properly distinguish wide and complex varieties of haptic feedback when focusing, we found that adding distractions in the form of talking and a roulette wheel drastically decreased a user’s confidence in the output of the device if there were too many motors. Furthermore, a user could not reliably distinguish between vibration levels on their foot. This limited the hardware to have two motors on the left and right of the foot to be easily distinguishable and only have the on and off states.

The output sector is encoded into binary to allow for fast and discernable feedback. Thus, if there are eight sectors to be predicted, the output requires two cycles of vibration, one for lower-order bits and one for higher-order bits. Testing found that around one second was the shortest time a user could read a cycle’s output and needed a short break of .2 seconds between cycles to distinguish them clearly. Additionally, a reset signal of 8 quick vibrations allows users to determine that the output is done and that the device is ready to gather more data for the next spin.

## Hardware

The hardware design focused much on the specific features new technology could afford us compared to Thorp and Shannon’s original design. Specifically, we sought to provide simplicity of design, adaptability to different persons, and ease of upgrade.

### Electrical

The main feature and improvement of the electricals is a microcontroller. With the first commercially available general-purpose processor launching in 1971 [4], years after Thorp and Shannon built their computer [6], the original gambling device did not have the chance to incorporate one in their design. Thus, we could greatly improve the upgradeability of our wearable device through software changes and create the potential of adjustability to other purposes, such as counting cards in blackjack.

Nothing has changed electrically since the device’s initial conception other than the switch from an accelerometer to a button for user input. The core of the device is an ESP32-C3 microcontroller running micropython firmware that all other components are connected to. Three general-purpose IO pins are used for the button and the two vibration motors; while the button is connected directly to the microcontroller, the vibration motors connect through a transistor whose source is the 3.3V line, and the gate is connected to the IO pin so they can draw more current. Lastly, a battery is connected to the 5V line to provide power to the microcontroller and the rest of the circuit.

### Mechanical

While the original device took the form of some boxes in the shoe [6], our device sought to utilize modern manufacturing processes to form the structure to be more organic and accessible, allowing multiple different users to comfortably utilize the device without the need to heavily modify their footwear.

The mechanical structure of the gambling device has gone through iteration of three designs until settling on the final version. Every time a design would be constructed, we would realize the shortcomings in the middle of the assembly process and immediately begin designing the next iteration.

The first iteration featured the device as a solid 3D-printed U-shaped object built to fit in the spaces around the foot within a shoe. The U shape minimized the need to expand a shoe to acquire extra space and also placed the vibration motors in the sections of the foot we found to be most easily deciphered distinctly and intuitively into a binary reading. While this design allowed for robust packaging, it failed to accommodate a variety of foot sizes and shoes. Furthermore, it could potentially cause discomfort due to its stiff structure. Finally, since the solid 3D-printed enclosure required modeling of organic structures that also needed to be accommodating, the plan was stopped in the early process of design.

Recognizing the limitations of a large, rigid design, we progressed to the next iteration where each distinct component would be surrounded by a moldable plastic material stuck onto a sock. The plastic was molded around a 3D-printed model of a generic 6-foot-tall person and cooled directly on the sock. This new design greatly enhanced comfort as it fit organically on the foot and expanded into the shoe naturally. Moreover, it was adaptable to a wide range of foot shapes and sizes, accommodating anything that could wear the sock, feet of American men’s size 7 up to size 12. Although the mold may not be specific to every foot, its fluid contours allowed it to still be comfortable despite any differences. Although the design worked well at first, issues emerged regarding lifetime, cleanliness, and uncomfortable interpersonal sharing of the sock prompting concerns about its permanent fixture to the fabric.

The final design separated the plastic from its direct contact with the sock, applying Velcro hook fasteners to the undersides of the molded enclosures [Figure 4.1]. When unhooked, the wires would keep the device together. The hooks would allow each component to be detached and reattached to any sock, adapting to an even wider variety of foot shapes and sizes and avoiding the obstacle of cleaning [Figure 4.2]. Although the fasteners did slightly decrease the grip strength of the device, testing revealed that it would remain in place as long as the foot was not aggressively thrust into a shoe. This final design managed to eliminate the weakness of the permanently affixed plastic and even improve its strengths of comfort and adaptability.

## Software

### 3.3.1 Algorithm

In the creation of our roulette prediction algorithm, we relied heavily on two main sources, each serving a distinct purpose in the development process. Edward Thorp’s original articles [5], were crucial in establishing a comprehensive understanding of the problem at hand. The second source [3] was more contemporary, playing a vital role in elucidating the physical model of the game of roulette. This resource was instrumental in laying the groundwork for our algorithm's implementation.

The creation of our roulette prediction algorithm began with converting mathematical equations into operational code. This task proved to be quite demanding, given our team's relatively modest background in this area of programming. This necessitated careful and precise efforts to transform these theoretical equations into practical, working code. We then moved on to prototyping prediction software components, which was crucial for debugging and understanding the results of each equation by allowing us to isolate and examine each part’s output. The subsequent integration of software and hardware was easier than expected due to clear documentation and a simple API. Here, we focused on testing and optimization for seamless functionality. In the final phase, we prioritized cleaning and refactoring the code to enhance efficiency, readability, and maintainability. This ensured that our software was optimized, efficient, and maintainable beyond being functional.

To better understand our project's success, it's essential to examine the inner mechanics of our roulette prediction algorithm

The input requires five distinct timestamps, each serving a unique purpose in the predictive process. These timestamps are used to measure three key splits: the time it takes the ball to complete two consecutive laps and the time it takes the wheel to complete a lap. Once the splits are accurately measured and the roulette's dimensions are known, the algorithm initiates its prediction process

The first part of the algorithm focuses on the roulette rim. Using the initial speed data, it calculates the velocity at which the ball will leave the rim. Additionally, it determines the elapsed time from the last recorded ball timestamp to this critical moment of departure.

With the calculated velocity at which the ball leaves the rim, the algorithm proceeds to compute the time it will take for the ball to descend to the level of the deflectors. This is achieved through time stepping, which incrementally evaluates the ball's trajectory and falling time, taking into account its speed and the physical properties of the roulette setup.

Finally, the algorithm uses the time duration from the last recorded timestamp of the ball until the estimated time when it hits the deflector. This duration allows us to compute the outcome in terms of the ball's position in radians relative to its location at the last timestamp. To output the result, we translate the result in radians to discrete sectors on the roulette wheel, letting the user bet within that sector before the table’s dealer stops bets from being placed.

### 3.3.2 Data collection and analysis

Our data collection and analysis process began with obtaining accurate measurements of the roulette wheel available to us. This initial step proved to be more challenging than anticipated, as precise and reliable measurement of the wheel's dimensions was essential for the accuracy of our predictions. The difficulty in obtaining these measurements could be a significant factor in the results we observed, which will be discussed in more detail later in our analysis.

Data collection emerged as one of the most time-consuming aspects of our project. Resorting to manual timestamp recording introduced noise due to the imprecision of hand-capturing points. Furthermore, data labeling required a meticulous review of slow-motion footage of the spins, demanding significant effort and heightening the possibility of human error, which could potentially influence the outcomes of our analysis and the performance of the algorithm.

Our data collection yielded three distinct datasets, each with its unique characteristics and challenges.

The first dataset was gathered without spinning the roulette wheel. This dataset's primary focus was on predicting the first deflector that the ball would hit. By eliminating the wheel's spin, we minimized noise and randomness post-deflection, providing a clearer assessment of the algorithm's predictive accuracy in a more controlled environment.

For the second dataset, we followed a similar approach but with a notable variation. Instead of focusing on the deflector as the outcome, we recorded where the ball landed on the stationary wheel. This method introduced a bit more noise compared to the first dataset but was still a significant step before tackling the complexities of a spinning wheel. This intermediary dataset helped us transition smoothly into scenarios involving more variables.

The final dataset mimics actual roulette conditions and introduces the highest level of noise, primarily due to the erratic bouncing of the ball. It is this dataset that would truly put our algorithm to the test.

In our datasets, we labeled outcomes in multiples of two to test the algorithm's accuracy across different sector sizes – halves, quarters, or eighths of the wheel. This approach helped us understand how the prediction accuracy varied with the granularity of the wheel's division. In analyzing our data, we used confusion matrices and key metrics like F1 score, accuracy, recall, and precision to evaluate our algorithm's performance. We also visualized the distribution of predicted and actual outcomes to better understand the data and identify any patterns or discrepancies.

# CHALLENGES

The goal of this project was to leverage foundational concepts of the more mature human-computer interaction field alongside improvements in technology to discover where we can improve the original gambling device. At the onset of this project, we identified five challenges and opportunities for exploration associated with this goal.

Motion detection was a challenge that was mostly avoided as a result of the change in the electrical and user interface design. While the original plan involved using an accelerometer for inputs, the switch to a single button simplified this process and allowed us to focus more on some of the other challenges that we thought were more integral to the success of the project.

Similarly, the concern over the accuracy and precision of the input largely came from the accelerometer. Prior experimentation with an accelerometer revealed differences in values obtained between different users, which would have caused some instances where input may be skipped or double counted. Using a button that mirrors Thorp and Shannon’s implementation of a microswitch allows us to partially rely on their determined timing errors of around .01 seconds [6].

The use of a simple button also simplifies the user interface to reduce cognitive load. Pressing a single button with the predefined intent of capturing timings simplifies the user flow. On the other hand, challenges arose in effectively conveying predictions through the foot, an unconventional interface of information. Extra affordances in terms of reset signals and breaks were necessary to keep the speed reasonable without confusing and overwhelming the user.

Anticipating real-time processing challenges, we initially chose a high-speed microcontroller with multithreading capabilities. However, during software development, we realized the speed of a single-threaded task was sufficient for rapid calculations. However, a new issue came up in the 2.2 seconds needed to relay the information to the user with haptic feedback. This delay accommodates the distractions in real-world usage. User training may reduce this time, but that may significantly increase the barrier to usage. Our implementation emphasizes the mental mapping of sectors to bets rather than a physical limitation of reading quick vibrations.

The largest challenge came from the data collection and validation of the final product. Initially, we looked into simulating the roulette wheel based on physics, but we found it would take more work and understanding than an algorithm to simplify it. Simulated data may also overlook factors such as varying friction on the wheel. We attempted to use computer vision to automatically collect data, but this failed due to time constraints. Thus, much of the data collection was done in a tedious, manual fashion, resulting in a much smaller data set than we wanted and potentially causing noise within the statistics. While labeled data provided some confidence in our outcomes, it limited incremental software design and validation over various roulette scenarios, given the inherent randomness and chaos of a roulette wheel.

# RESULTS

In our dataset focusing on predicting the deflector in roulette, we observed varying performance levels based on the number of sectors considered. With a sample size of 90 data points, the prediction accuracy for 2 sectors was slightly above 0.5, indicating a modest level of predictive capability. However, as the number of sectors increased to 4 and 8, there was a noticeable decline in performance. Specifically, the prediction accuracy dropped to approximately 0.15 for 4 sectors and further decreased to around 0.07 for 8 sectors. This trend suggests that the model's effectiveness diminishes significantly as the granularity of the prediction increases. [Figure 1.1]

The observed distribution of deflectors in our roulette dataset raises potential concerns about the inherent characteristics of our roulette wheel. Notably, deflectors 3, 5, and 7, which are the vertical deflectors, appear with significantly higher frequency. This suggests that these vertical deflectors are more likely to interact with and alter the path of the ball. Such a skewed distribution could be a key factor in explaining the marked decrease in prediction accuracy when the model considers all 8 deflectors. This discrepancy may be further intensified by the current approach in our algorithm, where the radius of the deflectors is averaged into a single number, without differentiating between vertical and horizontal deflectors. This lack of distinction could lead to an oversimplification, affecting the model's ability to accurately predict the final position of the ball [Figure 1.2].

Results from predicting the sector with a stationary roulette wheel showed promise despite the increased noise due to the ball's bounce. Achieving a higher prediction accuracy, particularly with 2 sectors, exceeds random chance expectations. However, similar to deflector prediction, accuracy falls when the prediction is extended to 4 and 8 sectors. This decline suggests that while eliminating the bias associated with deflectors improves overall predictability, the model still struggles with higher granularity in sector prediction, likely due to the complex dynamics of ball movement and bounce in a stationary wheel setup.

The dataset from a moving roulette wheel, although limited in size due to the time-consuming labeling, yielded notably high results, with an accuracy peak exceeding 0.6. This is impressive, considering the complexities involved in predicting outcomes with a dynamic, moving wheel. However, it's important to view this accuracy level as a preliminary result, given the inherent noise and randomness in roulette gameplay. To assess the algorithm’s true predictive power and ability to filter noise, a substantially larger dataset is needed to validate its robustness.

# IMPROVEMENTS

We have found five key areas to continue further development. Firstly, the algorithm could be refined with further discovery of the physics of the game to capture and resolve some of the randomness of the deflectors. Secondly, automating the labor-intensive data collection process through computer vision may significantly expedite the validation and iteration of the algorithm. Thirdly, the haptic feedback could be optimized or reworked to transmit the information faster, resulting in a less rushed experience when placing the bet. Fourthly, adding another layer of fabric or foam can reinforce the device’s structure while providing strain relief and mitigating disconnection risks when detached from a sock. Lastly, the device can be easily adapted to other casino games, such as blackjack, with simple modifications of the code while keeping the hardware the same.

# CONCLUSION

This paper’s roulette prediction device, while rudimentary, has shown a lot of promise. While much of the device follows the example set by its predecessors, the form-factor and vibration-based output methods make it stand out and add new ideas on how devices can silently and subtly communicate with their users. Namely, an added level of comfort and concealment with organic shapes and haptic rather than audio feedback makes the device a more pleasant and covert experience. While the code used for roulette prediction is not groundbreaking, it shows the viability of the device and provides a clear area for improvement for future development. The use of a microcontroller allows for a continuous cycle of future software upgrades that are easily integrated into the system. Further improvements could be identified through testing and prototyping, but the current state of the device has expanded upon Thorp and Shannon’s original work [5] and has demonstrated success at covertly predicting roulette outcomes.

## Reflection

We were initially ambitious in creating a device that would take motion input to predict outcomes in roulette and potentially expand that to other games like Blackjack. Even after simplifying the design, it felt like there was a constant rush to complete the cohesive system with many issues along the way. While hardware development went somewhat smoothly, the required in-depth understanding of the underlying physics was a major roadblock that backed algorithm development. Furthermore, data collection proved to be a much harder task than expected, shortening the available time for validation and continuous algorithm improvements. We managed to produce a product that performs and meets the expectations set at the onset of the project, but we kept building expectations and ideas for future improvement throughout the development process. Those further expectations could not be met given the lack of time, but we plan to continue to investigate and see how the vision of even greater odds and a more cohesive device can be brought to life.

After going through the undertaking of remaking the original roulette predictor, we came to see the conveniences of the components and concepts we used. Applying the variety of products and instruments that were available due to modern advancements in technology showed how rapid prototyping and continuous iteration were greatly eased. Stronger computing power, moldable materials, commoditized hardware, and smaller electronics all contributed to the actual possibility of constructing our device. Furthermore, the separation and parallelization of hardware and software development due to general-purpose microprocessors means that experts within independent fields can easily collaborate on projects that require a combination of their knowledge without massive interference between their work.

# REFERENCES

1. Roulette. J. Doyne Farmer. (n.d.). <https://www.doynefarmer.com/roulette>
2. Shoe computer. HNF. (n.d.). <https://www.hnf.de/en/permanent-exhibition/exhibition-areas/computers-for-everyone-1980-to-2000/hackers-reaching-out-for-forbidden-knowledge/shoe-computer.html>
3. M. Small and Chi Kong Tse, "Feasible implementation of a prediction algorithm for the game of roulette," APCCAS 2008 - 2008 IEEE Asia Pacific Conference on Circuits and Systems, Macao, China, 2008, pp. 1208-1211, doi: 10.1109/APCCAS.2008.4746243.
4. *The story of the Intel® 4004*. Intel. (n.d.). https://www.intel.com/content/www/us/en/history/museum-story-of-intel-4004.html
5. E.O. Thorp, “Physical Prediction of Roulette I, II, III, IV,” Gambling Times, May, July, August, October 1979.
6. E. O. Thorp, "The invention of the first wearable computer," Digest of Papers. Second International Symposium on Wearable Computers (Cat. No.98EX215), Pittsburgh, PA, USA, 1998, pp. 4-8, doi: 10.1109/ISWC.1998.729523.

# APPENDIX

## Predicting deflector

Figure 1.1

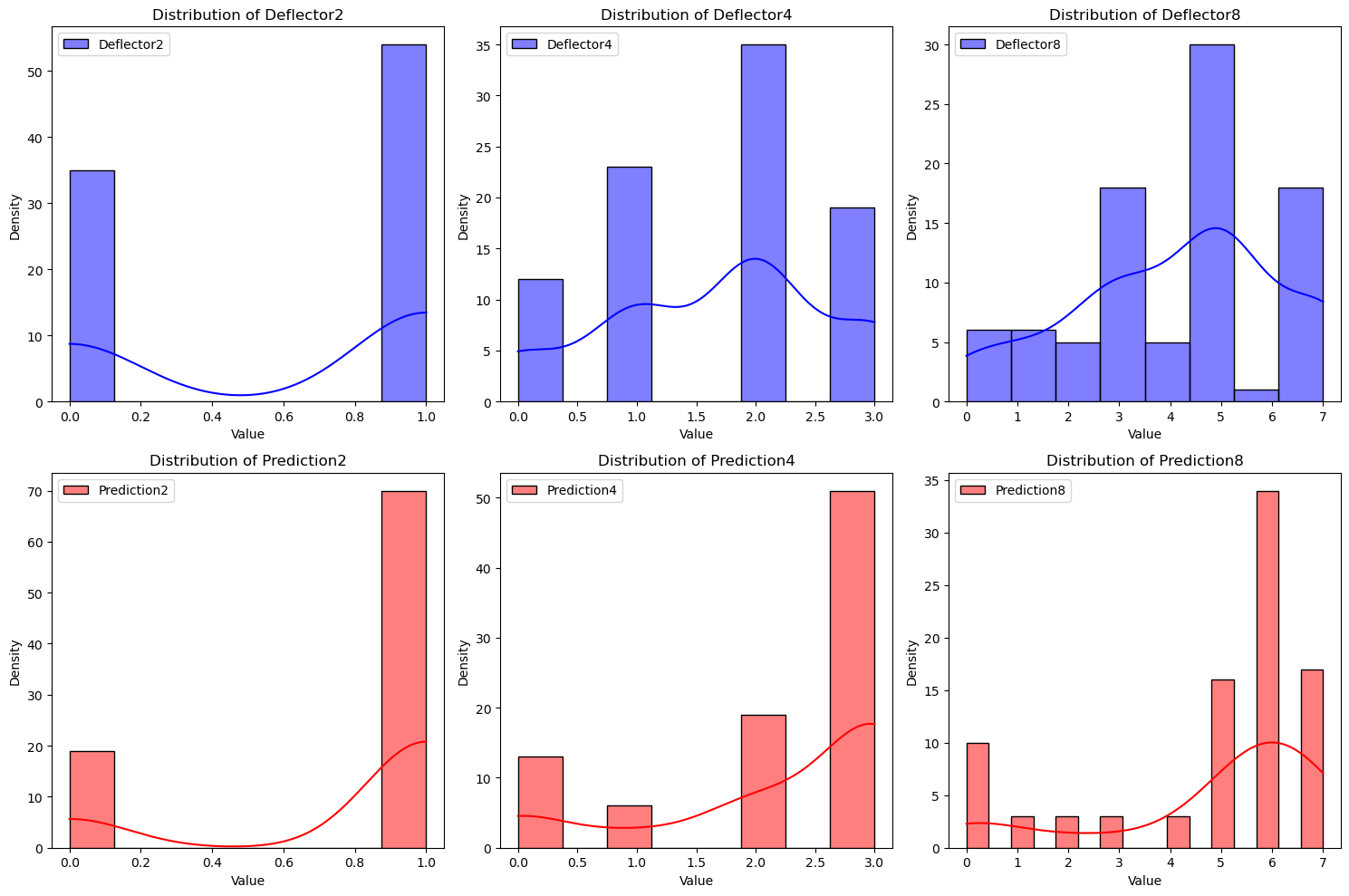


Figure 1.2

## Predicting with stationary wheel

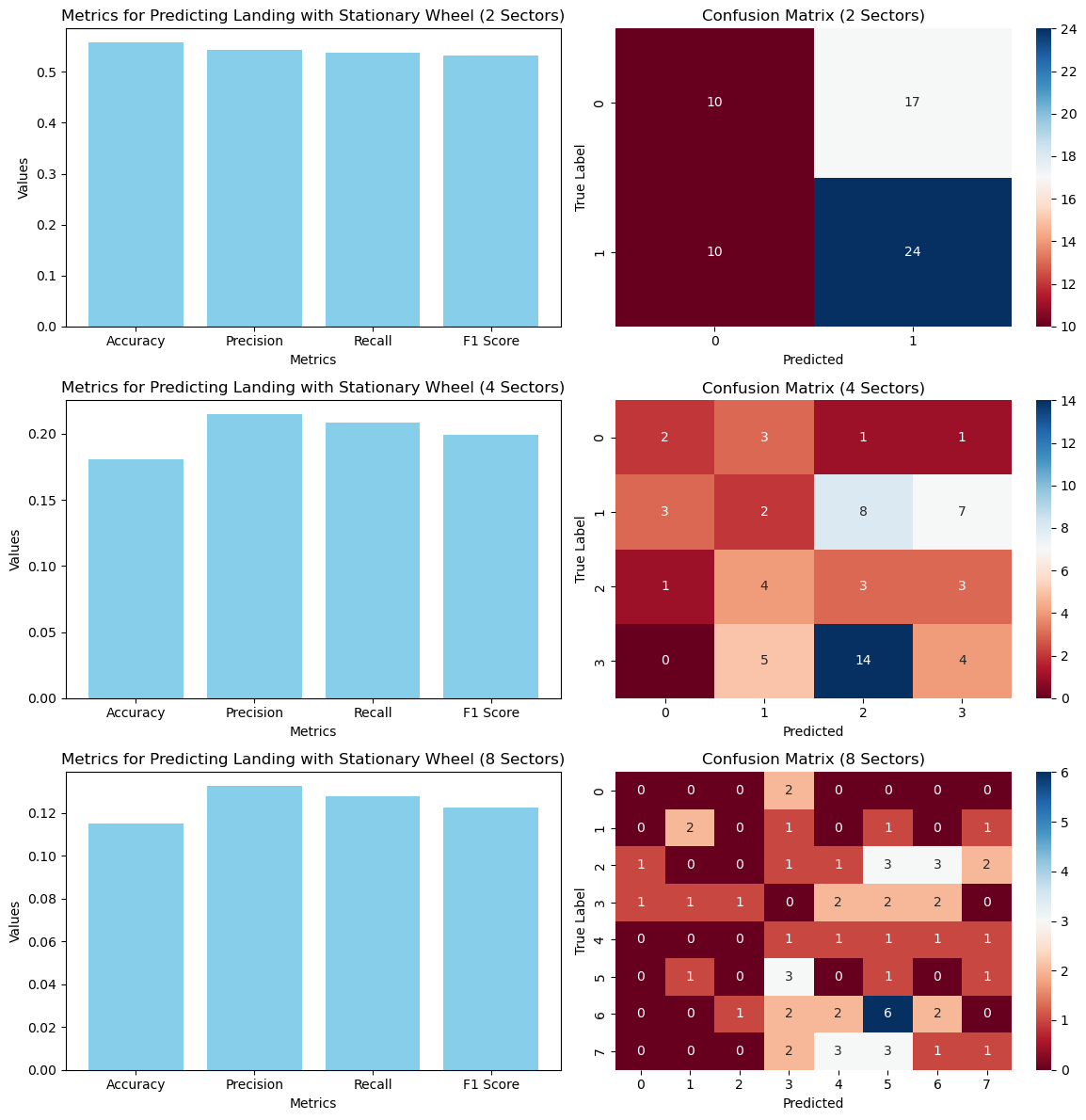


Figure 2.1

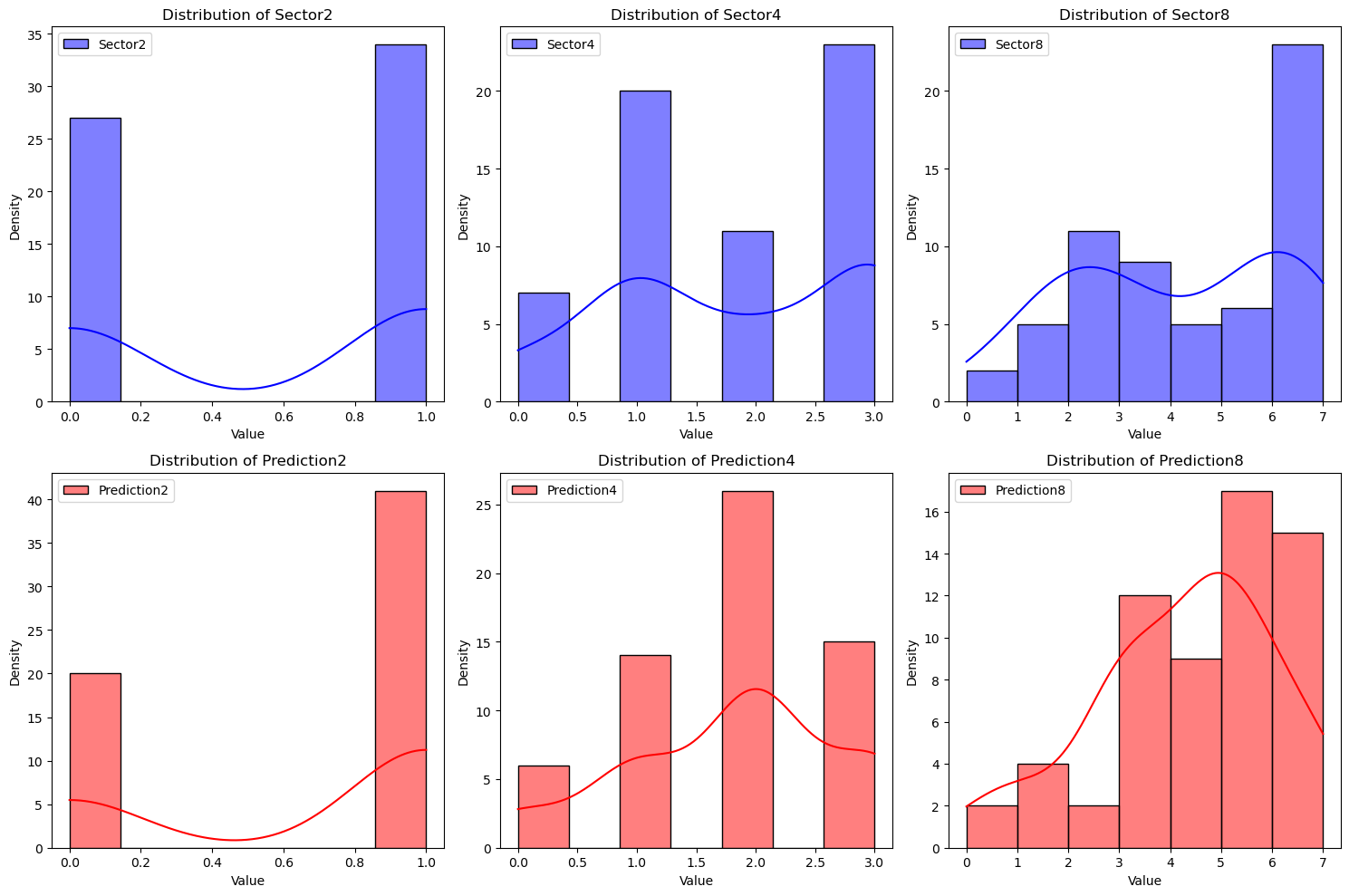


Figure 2.2

## Predicting with moving wheel

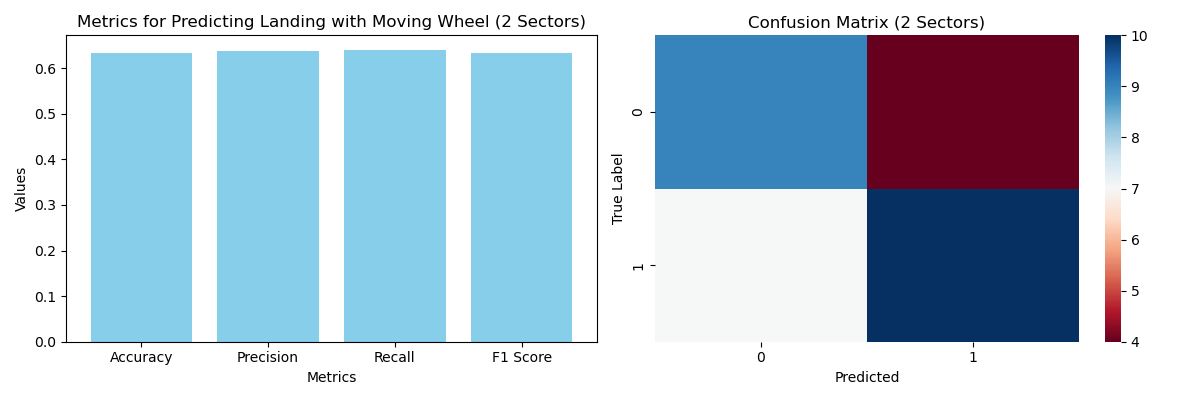


Figure 3.1

## Hardware



Figure 4.1

****

Figure 4.2