Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

THE USE OF SIMULATION WITH MACHINE LEARNING AND OPTIMIZATION FOR A DIGITAL TWIN- A CASE ON FORMULA 1 DSS

Andrew Greasley

Department of Engineering Systems and Supply
Chain Management,
College of Engineering and Physical Science,
Aston University,
Aston Triangle, Birmingham, B4 7ET,
UNITED KINGDOM.

Gajanan Panchal

Department of Operations and Information
Management,
College of Business and Social Science,
Aston University,
Aston Triangle, Birmingham, B4 7ET,
UNITED KINGDOM.

Avinash Samvedi

Department of Industrial Systems Engineering and
Management,
National University of Singapore,
1 Engineering Drive 2, Block E1A, #06-25, 117576
SINGAPORE.

ABSTRACT

The implementation of a digital twin presents a challenging environment for simulation. One challenge is the need for fast execution speed to maintain synchronization with the real system. When providing predictive outcomes, the complementary use of simulation with machine learning and optimization software may be employed to achieve this aim. The article investigates the use of simulation, machine learning and optimization in terms of providing a digital twin capability. The article presents a case on Formula1 or F1 competition, where a decision support system (DSS) framework is presented to explore a digital twin capability.

1 INTRODUCTION

The relationship between simulation and digital twins is close, with the first reference to the use of simulation as a digital twin (DT) published in 1993 (Katz and Manivannan, 1993). Since then, various terms have been used to describe the connection of simulation models with real systems and processes to support operational decisions. These include Near Real Time, Real Time, Cyber-Physical Systems, Semi-Physical Simulation and Symbiotic Simulation (dos Santos et al., 2021). This article provides a distinction between a simulation and a digital twin in terms of the level of data integration and organizational scope. This leads a review of the challenges from a simulation perspective of implementing a digital twin capability. One of these challenges is the need for synchronization of the simulation with real world data which highlights the need for fast execution of the simulation. Here the use of machine learning and optimization software is evaluated in their role as complementary components to simulation in providing capabilities such as fast execution required for a real time digital twin implementation. In this paper, we propose a decision support system (DSS) framework for a case of Formula 1 or F1. This framework integrates machine learning, optimization and agent-based simulation together with a digital twin.

978-1-6654-7661-4/22/$31.00 ©2022 IEEE

2 BACKGROUND

2.1 Defining a Digital Twin

Negri et al. (2019) define a digital twin as an integrated simulation of a complex product/system that, through physical models and sensor updates, mirrors the life of its corresponding twin. Tao et al. (2018) state that a digital twin consists of three parts: physical product, virtual product, and connected data that tie the physical and virtual product. Based on Tao et al. (2018) the following are characteristics of a digital twin:

- Real-time reflection. Two spaces exist in a digital twin, the physical space and the virtual space. The virtual space is the real reflection of the physical space, and it can keep ultra-high synchronization and fidelity with the physical space.
- Interaction and convergence. This characteristic can be explained from three aspects.
    - Interaction and convergence in the physical space. A digital twin is fully integrated so the data generated in various phases in the physical space can connect with each other.
    - Interaction and convergence between historical data and real-time data. Digital twin data not only depends on expert knowledge but also collects data from all deployed systems in real-time.
    - Interaction and convergence between physical space and virtual space. The physical space and virtual space are not isolated in the digital twin with connection channels between the two spaces.
- Self-evolution. A digital twin can update data in real time by comparing virtual space with physical space in parallel.

By combining the classification of digital twins by level of data integration and organizational scope we can see that the concept covers a wide range of applications (Figure 1)

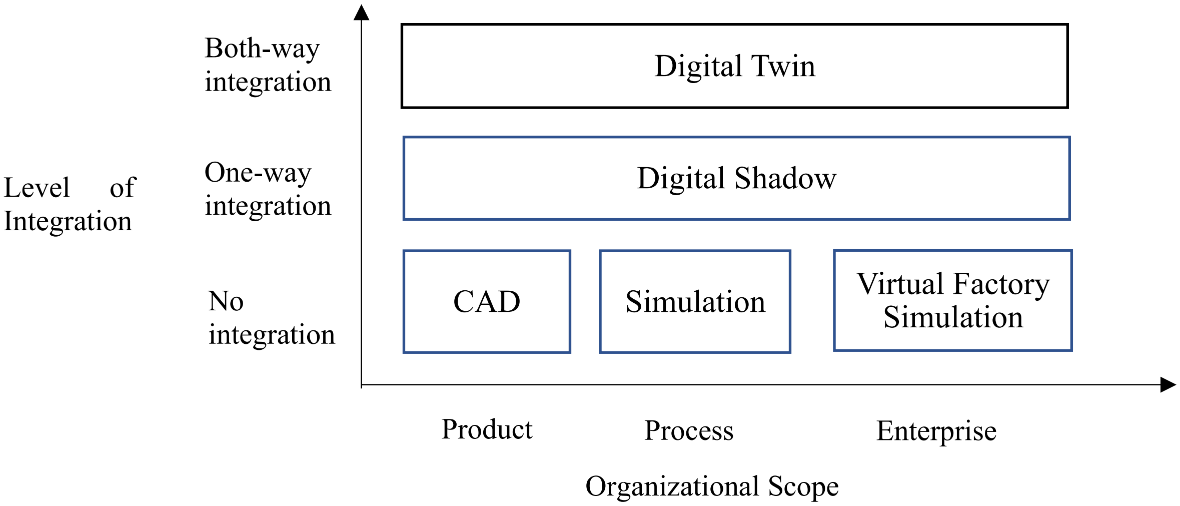

Figure 1: Level of data integration and organizational scope of a digital twin.

In the context of the organization, the scope of a digital twin can be at the product, process and enterprise level. At the product level this type of digital twins relates to the emulation of physical objects such as machines, vehicles, people, and energy. They can be considered as an extension of computer-aided design (CAD) and computer-aided engineering systems, which capture data that can then be used to detect issues and generate information that can be used to improve performance. They often have a focus on improving the efficiency of product life-cycle management, which is important for successful product-as-a-service or Servitization (Baines et al., 2017) business models. Digital twins allow monitoring of multiple products and resources in different operating conditions and different geographic locations.

At the process level this type of digital twin emulates processes over time and so require a dynamic simulation engine based on methods such as discrete-event simulation (DES). One application area is smart maintenance scheduling. An example is for predictive maintenance for a welding machine. Here a simulation provides a virtual representation in real time of the manufacturing process through data connections over the IoT. The current status of the welding machine is known by the digital twin. A machine-learning algorithm is used to provide a prediction of the remaining useful life of the manufacturing equipment based on its current usage and historical data of the process. The digital twin can be run into the future and predict machine failure based on its current status and scheduled future usage. The digital twin can then communicate back to the equipment to instigate a maintenance operation at the appropriate time. The digital twin thus provides an intelligent and automated predictive maintenance capability.

At the enterprise level, enterprise simulations aim to evaluate decisions made anywhere in the company on the performance of the whole business (Barton et al., 2001). Enterprise digital twins can be implemented by using multiple digital twins that are in use at the process level. Applications include connection of the digital twin to an enterprise resource-planning system in order to improve factory scheduling to reduce waste and management of inventory.

In terms of the level of data integration there are three possible levels of integration between the simulation and its real-world object counterpart (Table 1). When there is no automated data exchange between the simulation and the real-world object, when there is an automated one-way data flow from the real-world object which leads to a change in state of the simulation and when data flows fully integrated in both directions. These one-way digital twins may be referred to as Digital Shadows (Marquardt et al., 2021). Digital twins require a two-way data flow to provide a control capability to take action in response to predicted behavior. Corrective actions are often implemented using analytics methods based on machine-learning algorithms that provide appropriate methods of process control actuation.

Table 1: Level of integration of a digital twin.

ONGGO (2018)	**KRITZINGER (2018)**	**DESCRIPTION**
Type 1 (DSS)	No Way	A copy but not updated in real-time.
Type 2 (Control)	1 Way	Physical system sends data to the simulation. For type 1 a decision maker controls the physical system.
Type 3 (OpenLoop)	2 Way	Data flow in both directions. The simulation controls the physical system with an actuator.

In general, the level of complexity required of the simulation increases for a wider level of scope and for level of integration. The development of digital twins with fully integrated data flows in both directions is complex and is still in its infancy.

2.2 Simulation and Digital Twins

Before one looks at how do simulation and digital twins work in integration, it is important to understand them separately. Simulation can help understand what may happen in the real world. Digital twins not only help understand what may happen, but also what is happening in the real world (Taylor et al., 2021). Thus, simulation as digital twins are of most use when an object is changing over time, thus making the initial model of the object invalid, and when measurement data that can be correlated with this change can be captured (Wright and Davidson, 2020). In the guise of a digital twin this implies that simulation requires an ongoing capability in areas such as model updates, verification, validation and experimentation activities. Data flows between the simulation and real system may be in the following areas:

- Model realization data is required when the simulation model is defined and updated as a consequence of data from the real system. This is often referred to as a data-driven simulation. However, a purely data-driven model within a digital twin of a physical system is often not advisable as the data driven model is only reliable within the region of input parameter space from which the data used to construct the model was taken. Using data-driven models for extrapolation without imposing any constraints based on physical knowledge is a dangerous approach (Wright and Davidson, 2020).
- Model validation data is needed to ensure that changes in the real system are reflected in the simulation model. Checks for validity can be undertake at periodic intervals over time.
- Model experimentation data is needed in a sufficient quantity in order to update the simulation model parameters but if there are high levels of data processing then data reduction methods may be used in order to ensure timely processing is feasible.

Table 2 summarizes some of the challenges from a simulation perspective of implementing a digital twin capability and possible solutions.

Table 2: Simulation requirements for a digital twin application.

Simulation requirement	**Simulation solution**
Reactiveness (Fast prediction capability)	Distributed Simulation – Parallel or multiple simulation on a grid (network) and/or a cloud platform (Taylor et al., 2019) Multi-fidelity modelling (Cao et al., 2021) Combine with analytic methods (Li et al., 2021)
Real-Time Model Adaption	DDDAS – Adaptable Simulation Model (Fujimoto, 2019)
Interaction between the Physical and Simulated System	Symbiotic Simulation – Acquisition analysis of high volumes of data in real-time. Interface to physical system by decision maker or actuator. (Onggo, 2018).

In terms of fast prediction capability, three possible solutions are to implement a distributed simulation, undertake multi-fidelity modelling with different levels of granularity (Maier et al., 2017) or to combine simulation with analytic techniques such as machine learning and optimization. In terms of implementation of distributed simulation, Jain et al. (2017) state that the High-Level Architecture (HLA) (Kuhl et al., 1999) standard for distributed simulation, updated for web services support (IEEE, 2010) and the standard for COTS (commercial-off-the-shelf) Simulation Package Interoperability (SISO, 2010) are developments that have significantly facilitated the use of distributed simulation arrangements. Examples of cloud platforms that can facilitate rapid simulation execution include simulation packages such as Simio (https://www.simio.com/software/simio-portal.php) which uses the Microsoft Azure platform and Anylogic (https://www.anylogic.com/features/cloud/) which uses the Amazon Web Services platform. Taylor et al. (2019) discuss interoperability between models using identical simulation packages and between models using different simulation packages.

One requirement for a Digital Twin is the ability for real-time model adaption which is considered under the term Dynamic-Data-Driven Application Systems (DDDAS). An example of a DDDAS implemented using the Arena DES is provided in Celik et al. (2010). The implementation of adaptable data-driven models can be achieved through the use of a data-driven simulation approach (Goodall et al, 2019). This is primarily achieved by the definition of generic model objects with key data passed into the simulation from external files (Smith et al., 2018). Interaction between the physical and simulated system is considered under the term Symbiotic Simulation System (SSS) (Onggo et al., 2018). An SSS architecture proposes the use of simulation with big data analytics and data streaming technology.

This article will investigate the simulation requirements for a fast prediction capability. Here fast execution speed of the digital twin components is required because of the need for synchronization between the digital twin and the real system. Synchronization is important because when a relevant value changes in the real-world system but there is no (immediate) update in the simulation, the two systems drift apart. In such cases, automated updates to real-world systems based on digital twins may be systemically flawed (Marquardt et al., 2021). To enable a fast prediction capability, simulation can be combined with the analytic methods of machine learning and mathematical optimization algorithms to meet this challenge (Li et al., 2021). The following sections detail the Machine Learning and optimization approaches before we present the complementarities between the three approaches.

2.3 Machine Learning (ML)

Machine learning uses an iterative approach for the analysis of data in order to produce an analytical model. This model may be in the form of a mathematical equation, a rule set or an algorithm. Thus, machine learning does not refer to actual learning by a machine (computer) but the use of algorithms that through iteration provide an ability to predict outcomes from a data set. The main steps involved in machine learning are pre-processing of the data set, creation of a training set (usually 80% of the data) and a test set (usually 20% of the data) and selection of a learning algorithm to process the data. When compared to machine learning, simulation can be seen as the construction of a model that predicts outputs from inputs. Supervised learning looks at several input values with associated output values (termed labels), then through learning generates a model to predict outputs from inputs. Unsupervised learning looks at several input values of a random input value, then through learning generates a model in the form of a probability distribution that predicts outputs from inputs (Figure 2).

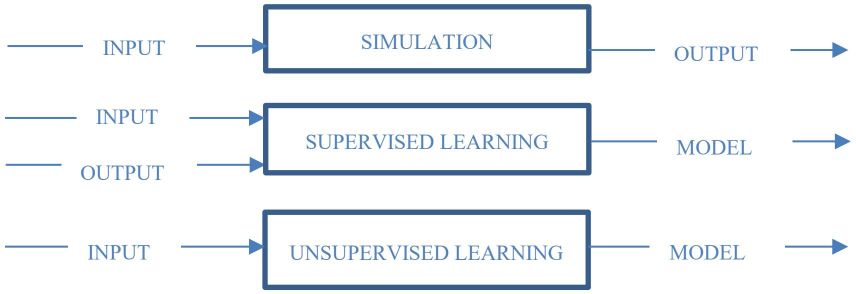

Figure 2: Simulation modelling, supervised learning, and unsupervised learning.

2.4 Optimization

Optimization involves finding the value of the input factors (levels) that provide the best outcome for a chosen output measure (response). Thus, optimization differs from ordering in that when ordering scenarios, we provide a number of given scenarios to compare against (usually no more than 20). In optimization we simply search through the possible alternative input levels to find the best scenario.

There are three main approaches to optimization.

2.4.1 Heuristics

In this 'manual' approach we use the domain knowledge of system experts to reduce the solution space and produce what we hope are optimum results. This approach involves running multiple experiments, with different experimental factor settings, to find the 'best' option. However, the approach may not find the optimal results, is only feasible for relatively simple optimization problems and is difficult to scale to complex issues.

2.4.2 Optimization Software

Many simulation software packages incorporate optimization software such as OptQuest (www.opttek.com) that use what can be termed a 'meta-heuristic' approach using; for example, the machine-learning technique of genetic algorithms. When optimizing using software the following four elements should be defined by the simulation user. Controls are the input parameters such as the capacities of resources that you define. Responses are the output parameters such as queuing statistics that you define. Constraints are the limitations imposed on the variables (controls and responses) such a minimum and maximum values. Objectives, sometimes referred to as the criterion function, is a statement of the goals of the system and how they are to be measured. (This may be expressed as the maximization of a variable such as profit level or minimization of a variable such as cost).

2.4.3 Reinforcement Learning (RL)

An alternative to optimization software is to use a Reinforcement Learning algorithm. This approach can handle a large solution space and multiple contradictory objectives. Traditional optimizers are static in that they optimize over a simulation run. A key advantage of the RL approach is that it can be triggered to dynamically optimize during the simulation run. The simulation can be used to train the algorithm and the optimized policy can then be adopted by the model (Greasley, 2020).

3 CONCEPTUAL FRAMEWORK

Figure 3 summarizes the complementary use of simulation, machine learning and optimization in the context of use in a digital twin.

Machine Learning has the advantage of fast execution of the ML algorithm and the ability to define outputs for decision making without knowing the real-world system mechanism. Bergmann et al. (2017) address the issue of how the decision rules derived by machine learning techniques can be used during the simulation run. The approaches suggested are to either call an external machine learning tool from the simulation system, or transfer or (re-)implement the decision model into the simulation system using its modelling and programming facilities. Bergmann et al. (2014) implement the first approach using an interface between the simulation and the Matlab Neural-Network Toolbox. The second approach is used by Bergmann et al. (2017) who translate a decision tree into nested "if- statements" that can be coded into the model. However, ML cannot easily predict out of its historical data set and requires high quality data and high-volume data for training and testing the ML algorithm. There are however, a number of challenges for the use of ML for prediction in an organizational context including the challenge of devising a learning algorithm and suitable training method. Another issue with ML models is the concept of interpretability. In the context of ML models, the term interpretability refers to how we can explain the predictions from these models. ML models are often associated with a lack of interpretability with the term 'black box' being used to indicate models that lack transparency and cannot be understood by humans. Not all ML models are 'black box' however and different ML model types have different level of interpretability (Morocho-Cayamcela et al., 2019). Generally neural networks have a low level of interpretability and represent non-linear and non-smooth relationships. Classifications rules have a high level of interpretability and represent linear and smooth relationships and are relatively easy to compute. Because the issue of the lack of transparency of black-box algorithms a discipline termed explainable artificial intelligence (XAI) has recently become popular that aims to make AI derived decisions reconstruct able and comprehensible ((Adadi and Berrada, 2018). (Feldkamp (2021) provides an overview of XAI methods and shows how these methods can be used for the output analysis of simulation data farming projects.

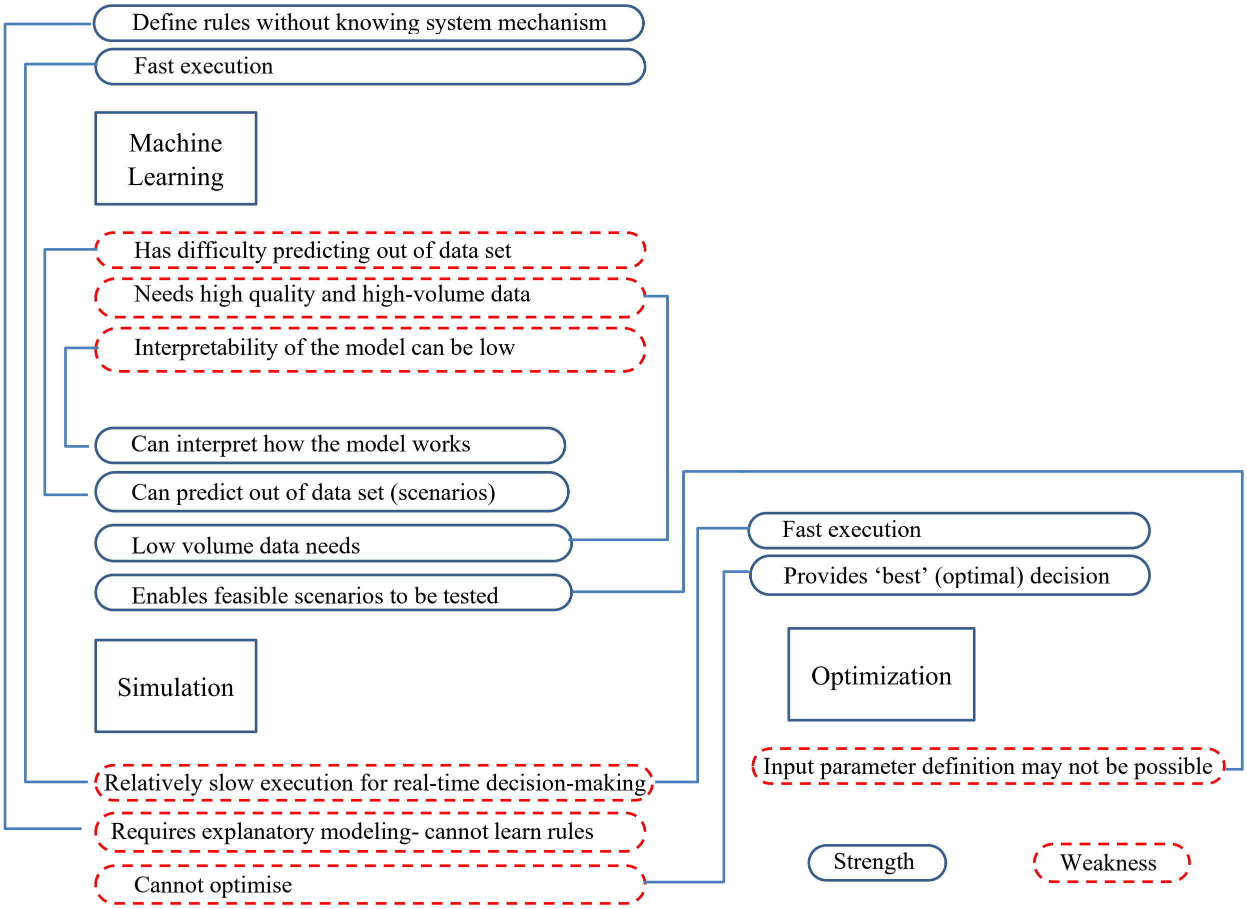
Define rules without knowing system mechanism
Fast execution
Machine Learning
Has difficulty predicting out of data set
Needs high quality and high-volume data
Interpretability of the model can be low
Can interpret how the model works
Can predict out of data set (scenarios)
Low volume data needs
Enables feasible scenarios to be tested
Simulation
Relatively slow execution for real-time decision-making
Requires explanatory modeling- cannot learn rules
Cannot optimise
Fast execution
Provides 'best' (optimal) decision
Optimization
Input parameter definition may not be possible
Strength
Weakness

Figure 3: The complementary use of machine learning, simulation, and optimization.

Optimization methods provide fast execution and will find the 'best' option from the solution space. There are however challenges when using optimization for prediction in an organizational. Although optimization approaches can be useful, in practice decision makers are concerned with providing what they consider to be a satisfactory answer to a problem. This ensures analysis is not overly extended in search of the "perfect" answer, but they are also aware that the model does not take into account all aspects of the real system. The limitations of the model mean that the optimum solution provided is only optimal in comparison with the other solutions tried and there are likely to be other solutions that exist that have not been considered by the optimization. Also, the decision maker is unlikely to make a decision solely on the basis of the model solution but is likely to consider other factors outside of the model scope to find a feasible and implementable solution. For example, an analysis to find an optimum work schedule by formulating a number of scenarios based on premises regarding ethical and moral concerns in terms of staff working conditions would be difficult to codify as parameters in an optimization algorithm.

Simulation models are constructed by simulation modelers and so with appropriate documentation including a statement of assumptions and simplifications, results from a simulation study should be understood derived from an understanding of how the simulation model works. We may find though that this causality is difficult to define for large and complex models. However, simulation has a relatively slow execution speed particularly for large complex models with complex experimentation needs. It also requires an exploratory model to be defined by the modeler and cannot find the optimum 'best' decision directly

4 EXAMPLE CASE STUDY- FORMULA 1 DSS

We use Formula1 or F1 competition case to demonstrate the usefulness of integrated framework of optimization with machine learning and simulation. This case is particularly suitable as it has a very complex decision-making process with a layer of performance under pressure. It demands a Decision Support System (DSS)-aided decision making under pressure. The F1 competition epitomizes business situations where time is critical and information systems cannot be separated from their context of use, neither in space nor in time (Heilmeier et al, 2018). As this racing competition is so complex with the incredible time pressure, there a whole team of experts who are feeding their analysis for winning strategies before and during the race as well. Each race demands about 5% of team's yearly budget in developing a high-performance DSS, which plays a key role in winning or losing a race. It also offers a very promising research area in strategic decision-making under pressure with a use of modern technologies.

4.1 Problem Definition

In F1, a pit stop is very critical where a F1 car pauses in the respective stalls (or pit) during a race for a quick maintenance, change of tires, mechanical repairs, and many other actions deemed necessary for winning the race in a safe manner. When to use a pit-stop and defining number of pit-stops are the most important decisions for the F1 team. This determines a success and failure in the race. During the race, the sensors fitted on the F1 car transmits the necessary data (i.e., race performance, activity of the subparts, and drivers' biophysical data etc.) for the team to analyze and meticulously plan a winning strategy almost instantaneously. The team runs through a complex simulation and predicts their relative position to other competing cars on the track and their best strategy under the racing conditions. The simulation models heavily rely on the historical data and team strategists' experience from previous race under the same racing conditions. The chief race strategist takes a look at the portfolio of strategies based on the simulation scenarios, consults with the race engineer, and makes a decision- such as pit stop, when to pit stop, or not pitting at all. Everything happens in a matter of a few minutes.

Fundamentally, there are three key questions that need to be answered as a race progresses:

1. When take a pit stop during the course of a race?
2. What tires to change during that pit stop?
3. What is the predicted track position after the pit stop and before the next pit stop?

As the competition is so intense with adversaries, an additional question that is useful to consider in the decision-making is:

1. What are the likely pit stop decisions made by the driver's rivals?

4.2 Proposed DSS

Here we define a DSS for F1 championship races. The methodology framework is shown in Figure 4. The methodology shows how supervised machine learning, reinforcement learning, optimization and simulation complement each other in finding the solution to such a complex real-life problem. Below we detail out the proposed methodology. First, we differentiate the solution based on time i.e. which steps of the solution are to be run before the race and which ones during the race. As can be seen from the figure, most steps are completed before the race. This helps in leaving only those steps which need live data and hence speeding up the live strategy recommendations.

4.2.1 Supervised Machine Learning

Each race generates massive data that requires a comprehensive analysis to come-up with various predictor models. The supervised machine learning helps unearthing various patterns from the historical data that assist the decision-making during the race. It also helps in making better race strategies for the future race in the similar circumstances. Although the human intelligence is important in winning the race, the machine learning enables a quick-thinking for the chief race strategist to execute the winning strategy. The decisions made are well structured and thus with the help of machine learning, the race engineers develop massive knowledge-base over the year.

There is a wealth of data which is generated from each race. The cars in F1 are among the most technologically advanced and have inbuilt sensors to collect a number of data points. This is done because even small changes in set-up can give significantly better lap times. Using this rich data, our first step is to find patterns, make associations and filter out models for various predictions needed. Some of them are; 1) Lap time predictor, 2) Tire degradation predictor, 3) Competition behaviour predictor, 4) Safety Car/Virtual Safety Car predictor

The above predictors and more are then input to an optimizer, which finds the best possible race strategies on pit stops and tires, but without taking the competition into account.

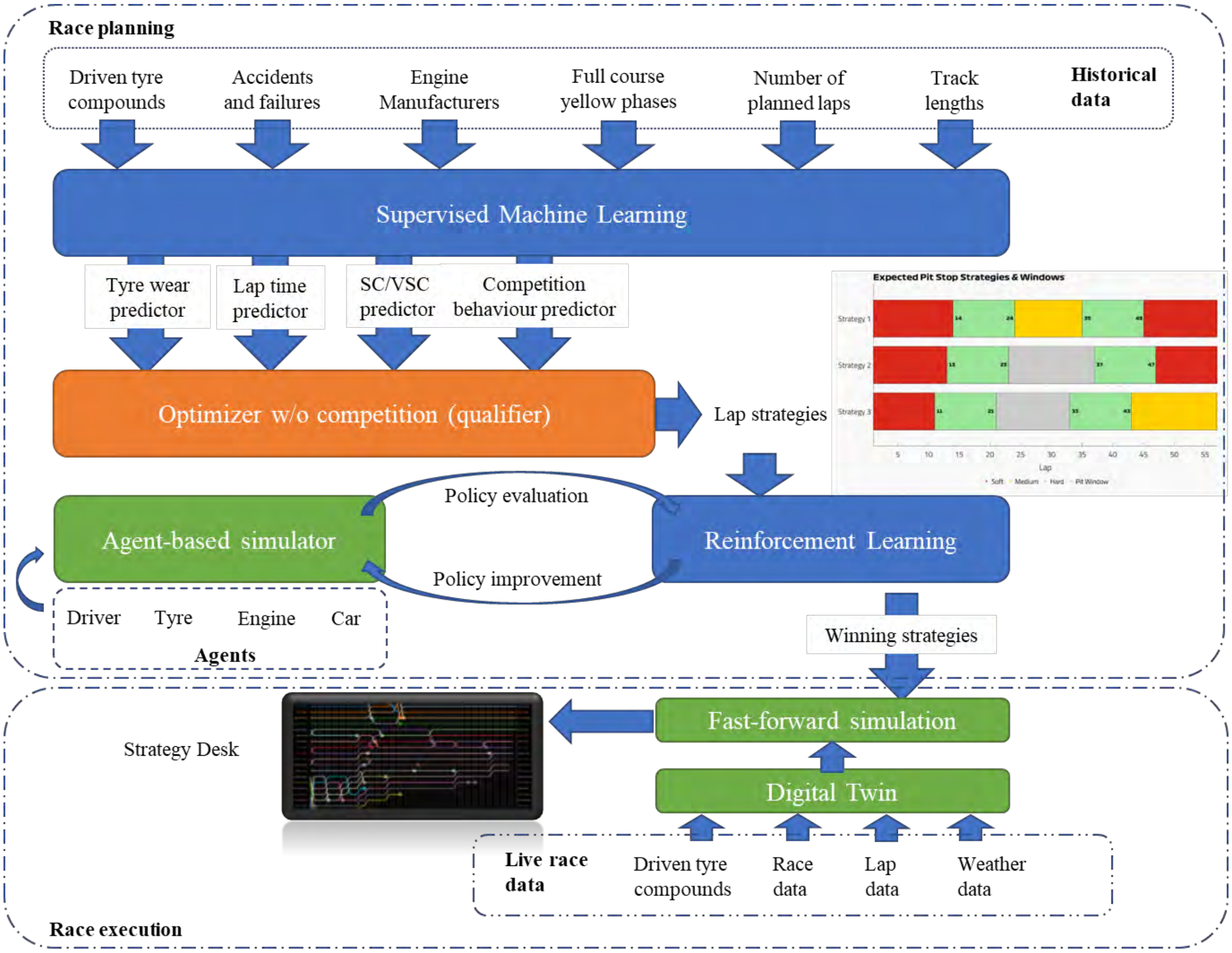

Figure 4: Formula 1 DSS framework.

4.2.2 Optimizer

Depending on the race, a number of pit stops are needed to replace worn-out tires. Assuming a race without opponents and considering only tire degradation, the optimal race strategy can be determined by solving a quadratic optimization problem. This is a minimisation problem where we have to minimise the total race time of the car. The total time can have multiple components as shown below:

$$t_{race,currentlap} = \sum_{lap=1}^{currentlap} t_{lap}(lap)$$

$$t_{lap}(lap) = t_{base} + t_{tire}(a_{tire}, c_{tire}) + t_{fuel}(lap) + t_{car} + t_{driver} + t_{grid}(lap) + t_{pit,inlap/outlap}(lap)$$

Where, t_{base} is the base lap time, t_{tire} is lap time loss due to tire degradation depending on tire age (a_{tire}) and rubber compound (c_{tire}), t_{fuel} is time loss due to fuel mass, t_{car} is time loss due to car abilities, t_{driver} is lap time loss due to driver's abilities, t_{grid} is first lap time loss due to starting grid position, $t_{pit,inlap/outlap}$ is lap time loss due to pit stops. This optimized solution (without competition) acts as the baseline of performance i.e. the best lap times possible. During actual race, multiple hindrances and inefficiencies will creep up and slow down the cars. These extra time additions can come from running in unclean exhaust air of the car ahead, problem in overtaking the car and hence reduced speed while following, inefficient pit stops (non-optimum pit window) etc. Hence to get a more realistic time frame we need to introduce competition and then optimize strategies against our competitor drivers.

4.2.3 Reinforcement Learning

With reinforcement learning, the strategy engineers assess their actions (decisions) and states (outcome). Fundamentally, the decisions are about when to pit stop by running various simulation scenarios on the results before the race as well as during the race. It is highly likely that the planned lap strategies need amendments based on various events, such as accidents that causes safety car (SC) or virtual safety car (VSC) deployment. Besides this, there are many incidences that affect the race in deriving variable lap times. Thus, it is important to understand and model these actions and states within the race simulation to anticipate the outcome of the race and put forward portfolio of strategies to select winning strategies.

Finding the optimal strategies, in real race setting requires making many sequential optimum decisions. Reinforcement learning is the best tool to handle this learning procedure. It requires input of action and state spaces and it then runs various simulations while taking different actions at states and learning the best actions to be taken at a state. For this problem:

- Action space: we consider as many pit-stop actions as the compounds available to the driver, with each of these actions representing a pit-stop to the respective compound, plus a stay on track action.
- State space: The state space is harder to model for this problem, as many factors must be considered to identify racing situations uniquely. The features that have been selected to describe the state of the race in the most compact way are: 1) Remaining laps, 2) Driver ranking, 3) Current tire-set, 4) Current tire-set age, 5) Exposed age, 6) Tire change rule already satisfied, 7) Remaining tire-sets, 8) Retired driver.

4.2.4 Agent Based Simulation

The above needs to be paired to a robust simulation model which is as close as possible to the real setting. This must be an agent-based model because each car needs to act independently and try to get the best strategy and win the race. There are a few agents considered in the agent-based simulator: driver, car, engine, tire, etc.

Reinforcement learning coupled with a realistic setting of an agent-based model then runs various races and try to find the best strategies to be used during the race.

During the race the digital twin will get the live data like location, tire condition, competitors' position, weather, competitors predicted strategies etc. and will connect to a simulator which quickly runs through a fast forward simulation to next few laps. This will help us:

- Get a quick look into the possible unfolding scenarios in next laps
- Collect statistical data and draw predictions
- Show these predictions through figures and tables

5 CONCLUSION

ML algorithms and simulation can be used in conjunction with optimization software to find best outcomes for real time or non-real time analysis. The true potentials of machine learning, optimization, and simulation are realized when integrated together. However, the limitations of both machine learning and optimization algorithms should be considered when combining these approaches. Further research should provide industrial case study evidence to quantify improvements in execution speed and provide a critical assessment of the combined use of these techniques. Achieving these benefits requires progress on a research agenda around the integration of data-driven and model-driven methods that ensures valid simulation models. For example, further research is needed to evaluate the use of approaches to embed machine learning algorithms in simulation models.

REFERENCES

Adadi, A. and Berrada, M. 2018. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI), IEEE Access, 6, 52138-52160.

Barton, J.A., Love, D.M., and Taylor, G.D. 2001. "Evaluating Design Implementation Strategies using Enterprise Simulation", *International Journal of Production Economics*, 72: 285-299.

Bergmann, S., Feldkamp, N., and Strassburger, S. 2017. "Emulation of Control Strategies Through Machine Learning in Manufacturing Simulations", *Journal of Simulation*, 11(1): 38-50.

Bergmann, S., Stelzer, S., and Strassburger, S. 2014. "On the use of Artificial Neural Networks in Simulation-Based Manufacturing Control", *Journal of Simulation*, 8(1): 76-90.

Cao, Y, Currie, C., Onggo, B.S., and Higgins, M. 2021. "Simulation Optimization for a Digital Twin using a Multi-Fidelity Framework", In *Proceedings of the 2021 Winter Simulation Conference,* edited by Sojung Kim, Ben Feng, Katy Smith, Sara Masoud, Zeyu Zheng, Claudia Szabo, and Margaret Loper, 1-12, Phoenix, Arizona: Institute of Electrical and Electronics Engineers, Inc.

Celik, N., Lee, S., Vasudevan, K., & Son, Y-J. 2010. "DDDAS-Based Multi-Fidelity Simulation Framework for Supply Chain Systems", *IIE Transactions*, 42(5): 325-341.

dos Santos, C.H., Montevechi, J.A.B., de Queiroz, J.A., de Carvalho Miranda, R., and Leal, F. 2021. "Decision Support in Productive Processes Through DES and ABS in the Digital Twin Era: A Systematic Literature Review", *International Journal of Production Research*, 60(8): 2662-2681.

Feldkamp, N. 2021. "Data Farming Output Analysis using Explainable AI", In *Proceedings of the 2021 Winter Simulation Conference,* edited by Sojung Kim, Ben Feng, Katy Smith, Sara Masoud, Zeyu Zheng, Claudia Szabo, and Margaret Loper, 1-12, Phoenix, Arizona: Institute of Electrical and Electronics Engineers, Inc.

Fujimoto, R., Barjis, J., Blasch, E., Cai, W., Jin, D., Lee, S., and Son, Y-J. 2018. "Dynamic Data Application Systems: Research Challenges and Opportunities", *In proceedings of the 2018 Winter Simulation Conference*, edited by Markus Rabe, Angel A. Juan, Navonil Mustafee, Anders Skoogh, Sanjay Jain, and Björn Johansson, 664-678, Gothenburg, Sweden: Institute of Electrical and Electronics Engineers, Inc.

Greasley, A. 2020. "A Simulation of Autonomous Robot Movement Directed by Reinforcement Learning", *In proceedings of 32nd European Modeling and Simulation Symposium, EMSS 2020*, edited by Affenzeller, M., Bruzzone, A. G., Longo, F. & Petrillo, A, 10-15. Creative Commons Attribution.

Heilmeier, A., Graf, M., and Leinkamp, M. 2018. "A Race Simulation for Strategy Decisions in Circuit Motorsports". *In proceedings of 21st International Conference on Intelligent Transportation Systems (ITSC),* 2986-2993. November 4th-7th, Maui, Hawaii .

IEEE (2010). 1516.1-2010 IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Federate Interface Specification, NJ: Institute of Electrical and Electronics Engineers.

Jain, S., Shao, G., and Shin, S-J. 2017. "Manufacturing Data Analytics using a Virtual Factory Representation", *International Journal of Production Research*, 55(18): 5450-5464.

Katz, D., and S. Manivannan. 1993. "Exception Management on a Shop Floor using Online Simulation", *In Proceedings of the 1993 Winter Simulation Conference*, edited by G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles. 888-896. Los Angeles, CA.

Kritzinger, W., Karner, M., Traar, G., Henjes, and J., Sihn, W. 2018. "Digital Twin in Manufacturing: A categorical literature review and classification", *IFAC PapersOnLine,* 1016-1022.

Kuhl, F., Weatherly, R., and Dahmann, J. 1999. "*Creating Computer Simulations: An Introduction to the High Level Architecture*", *NJ*: Prentice Hall.

Li, H., Cao, X., Jin, X., Lee, L.H., and Chew, E.P. 2021. "Three Carriages Driving the Development of Intelligent Digital Twins – Simulation Plus Optimization and Learning", In *Proceedings of the 2021 Winter Simulation Conference,* edited by Sojung Kim, Ben Feng, Katy Smith, Sara Masoud, Zeyu Zheng, Claudia Szabo, and Margaret Loper, 1-12, Phoenix, Arizona: Institute of Electrical and Electronics Engineers, Inc.

Marquardt, T., Morgan, L., and Cleophas, C. 2021. "Indolence is Fatal: Research Opportunities in Designing Digital Shadows and Twins for Decision Support", In *Proceedings of the 2021 Winter Simulation Conference,* edited by Sojung Kim, Ben Feng, Katy Smith, Sara Masoud, Zeyu Zheng, Claudia Szabo, and Margaret Loper, 1-12, Phoenix, Arizona: Institute of Electrical and Electronics Engineers, Inc.

Morocho-Cayamcela, M.E., Lee, H., and Lim, W. 2019. "Machine Learning for 5G/B5G Mobile and Wireless Communications: Potential, Limitations, and Future Directions", *IEEE Access*, 7, 137184-137206.

Negri, E., Fumagalli, L., Cimino, C., and Macchi, M. 2019. "FMU-supported Simulation for CPS Digital Twin", *International Conference on Changeable, Agile, Reconfigurable and Virtual Production*, Procedia Manufacturing, 28, 201-206.

Onggo, B.S., Mustafee, N., Juan, A.A., Molloy, O., and Smart, A. 2018. "Symbiotic Simulation System: Hybrid Systems Model meets Big Data Analytics", *In proceedings of the 2018 Winter Simulation Conference, edited by* Markus Rabe, Angel A. Juan, Navonil Mustafee, Anders Skoogh, Sanjay Jain, and Björn Johansson, 1358-1369, Gothenburg, Sweden: Institute of Electrical and Electronics Engineers, Inc.

Smith, J.S., Sturrock, D.T., and Kelton, W.D. 2018. *Simio and Simulation: Modeling, Analysis, Applications*, 5th Edition, South Caroline, USA, Simio LLC.

Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., and Sui, F. 2018. "Digital Twin-Driven Product Design, Manufacturing and Service with Big Data", *The International Journal of Advanced Manufacturing Technology*, 94: 3563–3576.

Taylor, S.J.E., Johansson, B., Jeon, S., Lee, L.H., Lendermann, and P., Shao, G. 2021. "Using Simulation and Digital Twins to Innovate: Are We Getting Smarter?", In *Proceedings of the 2021 Winter Simulation Conference, edited by* Sojung Kim, Ben Feng, Katy Smith, Sara Masoud, Zeyu Zheng, Claudia Szabo, and Margaret Loper, 1-12, Phoenix, Arizona: Institute of Electrical and Electronics Engineers, Inc.

Taylor, S.J.E. 2019. "Distributed Simulation: State-of-The-Art and Potential for Operational Research", *European Journal of Operational Research*, 273: 1-19.

Taylor, S.J.E., Mustafee, N., Turner, S.J., Pan, K., and Strassburger, S. 2009. "Commercial-Off-The-Shelf Simulation Package Interoperability: Issues and Futures", *In Proceedings of the 2009 Winter Simulation Conference*, edited by *M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls*, 203-215, Austin, Texas: Institute of Electrical and Electronics Engineers, Inc.

Wright, L. and Davidson, S. 2020. "How To Tell the Difference Between a Model and a Digital Twin", *Advanced Modeling and Simulation in Engineering Sciences*, 7(13): 1-13.

AUTHOR BIOGRAPHIES

ANDREW GREASLEY is a Lecturer at Department of Engineering Systems and Supply Chain Management College of Engineering and Physical Science, Aston University, Birmingham, UK. He has taught in the UK, Europe, and Africa at a number of institutions. Dr. Greasley has over 100 publications with 13 books including Operations Management, Wiley; Operations Management: Sage Course Companion, Sage; Simulation Modeling for Business, Ashgate; Business Information Systems, Pearson Educations (Co-author); and Enabling a Simulation Capability in the Organization, Springer Verlag. He has provided a simulation modeling consultancy service for 30 years to a number of companies in the public and private sectors including ABB Transportation Ltd. (now Bombardier), Derbyshire Constabulary, GMT Hunslet Ltd., Golden Wonder Ltd., Hearth Woodcraft Ltd., Luxfer Gas Cylinders Ltd., Pall-Ex Holdings Ltd., Rolls-Royce Ltd. (Industrial Power Group), Stanton Valves Ltd., Tecquipment Ltd., Textured Jersey Ltd., and Warwickshire Police. His e-mail address is a.greasley@aston.ac.uk.

GAJANAN PANCHAL is a Lecturer in Operations Management and Supply Chain Management at Aston Business School, Birmingham, UK. Before joining Aston Business School in 2019, Dr. Panchal has about six years of research experience in the area of supply chain and logistics management at the National University of Singapore. He holds a Ph.D. degree in Industrial Engineering from Indian Institute of Technology Delhi, India. Her research interests include Supply chain design, Mechanism design, Servitization, Digital twinning for supply chain and logistics operations. His email address is g.panchal@aston.ac.uk.

AVINASH SAMVEDI is a Senior Research Fellow at the Centre of Excellence in Modelling and Simulation for Next Generation Ports (C4NGP) of Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore. He obtained his PhD from Indian Institute of Technology Delhi in 2012. Post PhD he has five years of research experience and a further three years experience in a leading logistics industry at Singapore. His research interests include supply chain analytics, with emphasis on simulation and optimization, applied to supply chain risk management, vehicle routing and inventory management. His email address is iseavi@nus.edu.sg.