HIGH SCHOOL EDITION
Journal of Student Research
Volume 13 Issue 4 (2024)

A Systematic Review and Analysis of Machine Learning Models in Liquid Rocket Engine Control

Malhar Gandhe

Videsh Team High School, India

ABSTRACT

This paper explores machine learning (ML) methods that enhance performance, design, health, and operation of liquid rocket engines. Various ML approaches, including reinforcement learning (RL), supervised, and unsupervised learning can potentially transform rocket propulsion technologies, essential for critical interplanetary missions. Specifically, this study reviews neural network-based models for health monitoring of rocket engines, RL for control of engine ignition and operation, and ML techniques for anomaly detection. The application of these algorithms leads to significant advances in analyzing and predicting rocket engine system performance. While such techniques enhance efficiency, they also present challenges, which are discussed herein. Subsequently, this study provides a comprehensive synthesis of ML techniques as applied to rocket engine diagnostics and prognostics. This approach not only addresses the existing research gap but also lays the groundwork for future explorations into the predictive maintenance and optimization of space propulsion systems.

Introduction

Liquid rocket engines are complicated systems with high performance needs that require testing to meet the criteria for space exploration and satellite deployment missions [1, 2]. Traditional approaches to engine design and testing rely heavily on empirical data and extensive physical testing, which are both time-consuming and costly [3, 4]. Recent advancements in machine learning provide new opportunities to enhance the efficiency, safety, and reliability of these engines through sophisticated predictive models and control systems [5, 6].

The use of machine learning in rockets is beneficial in many ways. In this paper, we examine liquid rocket propulsion (LRP), which is preferred over solid rocket propulsion (SRP) due to its higher specific impulse [7]. LRP can be made reusable. This has been experimented in recent years. Controllability is another advantage in LRP systems. Although LRP systems have numerous advantages, complexity remains an unresolved issue. However, recent advancements in machine learning methods show potential for minimizing this complexity, making LRP systems easier to develop. [5]. Over the last decade there has been a substantial increase in achieving key breakthroughs with the use of machine learning techniques for liquid rocket propulsion [8]. This is due to the vast data humans have generated over the years in this field. A major source is Space shuttle main engine (SSME) data which is used by many to understand and evaluate the behavior of ML programs [9, 10,11]. Many attempts have been made to improve the feasibility for supporting the operation of rocket engines using neural networks [4]. Numerous algorithms such as reinforcement learning, supervised, unsupervised learning algorithms have been used to predict results from the test data. Also, there are many software which have been used to detect certain anomalies in rocket engines, couple of which are mentioned further in the paper. [10, 11].

The development and operation of liquid rocket engines involve challenges like thermal management, material fatigue under extreme conditions, and dynamic control during various phases of flight [12]. Addressing these issues using traditional methods is increasingly unfeasible due to the rising complexity and cost constraints in space missions. Thus, there is a pressing need for innovative approaches that can provide quick, reliable, and cost-effective solutions. The potential of ML algorithms can be utilized in four areas of liquid rocket engines. First, to optimize combustion by providing vast amounts of sensor data from engine tests to these ML algorithms [4]. By analyzing this data, ML models can predict potential in-flight issues and can also ensure ideal engine performance. Second, ML can evaluate optimum configurations for combustion chambers, fuel/oxidizer injectors and cooling systems for nozzles [38]. Third, these algorithms can also be used to detect anomalies in real time during flight to avoid any catastrophic failures [13]. And lastly to operate control systems, ML algorithm-controlled system can analyze and adjust parameters like fuel flow rate or oxidizer/fuel ratio to adapt to necessary conditions.

Machine learning algorithms offer many advantages to the space propulsion industry. Their robustness is crucial for this industry. A concern that arises is whether the ML software can improve classical simulations in a way that reduces operational costs or enhances calculated accuracy. The answer is yes in this case; algorithms such as Gaussian processes, random forests and neural networks can be trained to build models which can approximate many behaviors in propulsion such as complex fluid flow or thermal analysis in nozzles [14, 9]. This review paper's objective is to summarize and analyze different ML methods like reinforcement learning, supervised learning, unsupervised learning and also algorithms such as Random forests, neural networks, Gaussian models with the help of software like ORCA and GritBot to analyze liquid rocket engine control.

This review paper will be valuable to the research community by providing a comprehensive overview of the current machine learning methods applied in predicting and controlling the performance of liquid rocket engines, thereby facilitating informed decision-making and guiding future research in this specialized area. It highlights existing approaches, offering insights into their effectiveness and potential areas for improvement.

The paper starts with an introduction. We then analyze compare the different ML frameworks. Subsequently, we discuss and present our findings before ending the paper.

Methodologies

Machine Learning as an Overview

The term machine learning was coined in 1950's with the development of various mathematical models required for developing neural networks. Machine learning (ML) became very useful in many industries and was quickly adopted in the field of rockets as well. Today ML algorithms are one of the many ways and the most effective ways in analyzing anomalies present in rocket propulsion systems. ML algorithms can analyze sensor data from rocket engines such as pressure, temperature, fuel flow et cetera which can help in developing the design and optimizing rocket nozzles as well as engines. Fig.1 demonstrates various ML algorithms which are discussed in this section.

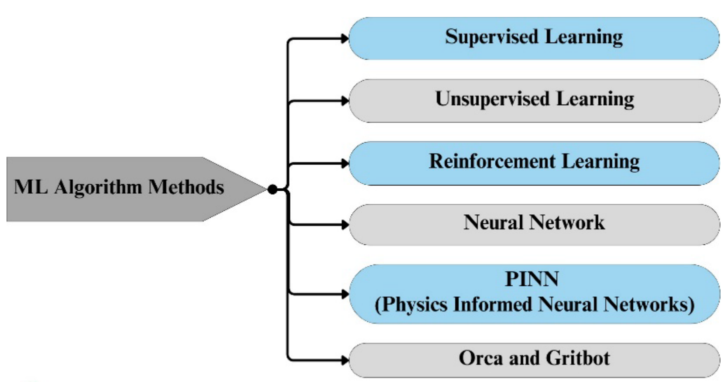


Figure 1. Various algorithms used in liquid rocket propulsion systems.

We have captured vast data by performing numerous engine tests and rocket launches in the last couple of decades. This data when used by ML algorithms make the complex rocket engine systems more effective. In this section we will discuss various ML algorithms such as Supervised Learning, Semi-Supervised Learning, Unsupervised Learning, Reinforcement Learning, Neural Network, PINN and software such as ORCA and GritBot which detect anomalies.

Supervised Learning

In the last decade, a large number of supervised learning methods have been introduced in the field of the machine learning [15]. Supervised Machine Learning generates the desired output and makes a prediction based on the trained dataset provided in the input [16]. Various algorithms under Supervised ML including Naïve Bayes, Logistic Regression, Random Forest, J48, CART, Multi-Layer Perceptron, Support Vector Machine (SVM) which are common and famously used by researchers [16].

• Naïve Bayes: A Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature [17].

• Support Vector Machines (SVM): In machine learning, support-vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis [17].

• Random Forests: Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [18].

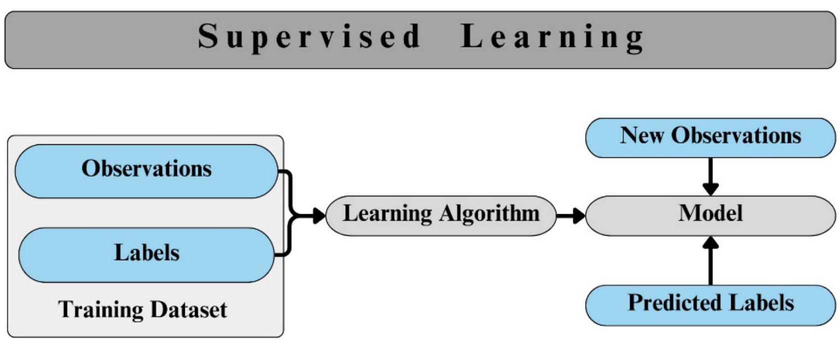


Figure 2. A flowchart depicting how a supervised learning algorithm works.

Supervised learning algorithms are training dataset which contain both the inputs and the desired outputs, and the goal is to learn the corresponding mapping rule [4]. Once this mapping rule is recognized and understood it can make the performance of a liquid engine more powerful, for example, when a supervised learning algorithm analyses data on heat spots in nozzles, necessary cooling areas, pressure indexes near nozzle throat areas, fuel injectors configurations, the algorithm can then make the nozzle geometries much more efficient. Such designs made by the algorithms may predict combustion instabilities during flight [4], may detect expected anomalies by seeing the pattern of the data provided. To ensure ideal performance of rocket engine, it is very necessary to enable high temperatures through a regeneratively cooled nozzle[19]. To enable high temperatures there is a need to protect the solid surfaces exposed in a high-temperature environment, introduction of film cooling in the combustion chamber not only protects the wall from high thermal loads, but also from chemical impact [20]. Evaluation and selection of materials in the areas of high temperature exposure, is also very important to ensure optimal performance [21].

Unsupervised Learning

Unsupervised learning, a branch of machine learning, aims to discover structures, relationships, or patterns in data without the aid of labels or other target variables [22]. In Unsupervised learning the results are not predetermined, and the algorithm learns and analyses the data to predict the output based on the patterns already present in the provided data. The steps that are involved in this program are as follows:

• Data collection (clustering): As unsupervised learning deals with unlabeled data, there is a need to pre-process the data in order to avoid any inconsistencies within the data, inconsistencies may include missing values, incorrect data et cetera.

• Learning Algorithm: The algorithm must be trained on the data for it to predict results by analyzing patterns, common choices include K-means clustering for grouping data, PCA for dimensionality reduction, and isolation forests for anomaly detection.

• Validation: After the model is trained on a pre-processed data, it is time to evaluate/validate the model. Internal strategies evaluate the goodness of a model without using any external information [28]. If the results are not ideal, we would have to go back to step 2 (Learning) to select a different model.

• Model: Lastly, this is the step where all the data is grouped, data is organized according to its features and configuration and the anomalies detected are recorded.

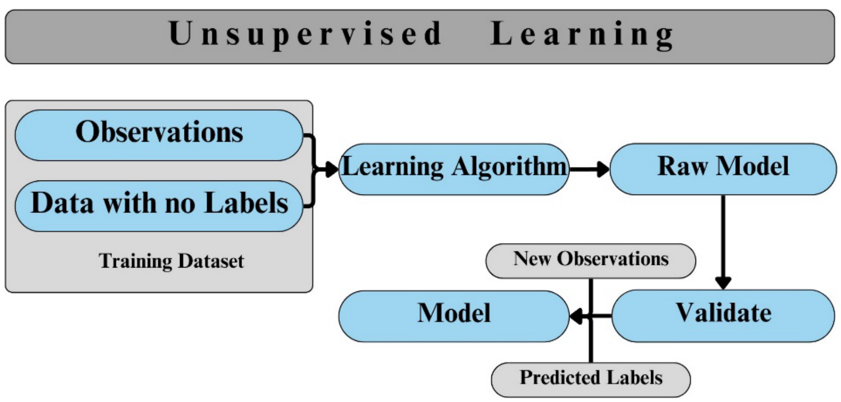


Figure 3. A flowchart depicting how an unsupervised learning algorithm works.

Unsupervised learning algorithms are going to be very useful especially in the space propulsion industry. For example, these algorithms can be used to predict trajectories for a rocket to land back on ground, following that it can also be very useful in retro-thrusting, if we provide sufficient parameters in the data such as velocity of rocket while landing, coordinates for the rocket to land, mass flow rate of the propulsion system, et cetera, the algorithm may come up with a pattern which would retro-thrust the rocket while landing.

Reinforcement Learning

Reinforcement learning is the problem faced by an agent that learns behavior through trial-and-error interactions with a dynamic environment [26]. For example, a hungry dog would get a treat (that is a reward) for fetching the ball (action), learning to repeat that particular task. This algorithm of a reward for a desired action is depicted in the Fig. 4 below.

In the case of rockets training such algorithms can be expensive and may even not be feasible at all for liquid rocket engines due to safety concerns [13]. An alternative is to train the controller in a simulated environment and transfer the learned policy afterwards to the on-board embedded computer of the rocket engine [13]. The use of this algorithm can be specifically seen in health monitoring of rocket engines. With the help of Markov Decision processes (MDP) [40] the task of health monitoring can be broken down into 5 components:

• States: This section represents health parameters such as engine temp, engine pressure, mass flow rate et cetera.

• Actions: This step would process system checks, adjust any controls, or abort the system test if needed.

• Transition Probabilities: Models the likeliness of moving from one state to another given an action. For example: Performing a specific check on a system might have a high probability of revealing an issue.

• Rewards: If any issue persists and is solved, the rocket engine would perform much better than earlier, this can be thought of a reward.

• Policy: A policy specifies the best action to take in each state to maximize the expected reward.

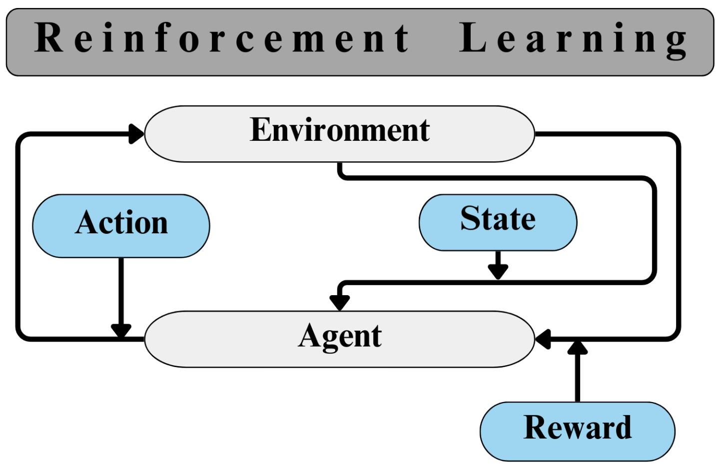


Figure 4. A flowchart depicting how a reinforcement learning algorithm works.

These reinforcement algorithms are currently being used in health monitoring of liquid fueled rocket engines and to perform soft landings back on earth. Reinforcement Learning (RL) allows to train a neural network controller for the rocket engine and enables to deduce an optimal flight path for different mission scenarios by interacting with suitable simulation environments [27].

Neural Network

Neural networks have received much attention in engineering applications in the last decade because they are highly flexible and have the ability to be trained, using user-supplied data, to map complex surfaces [36]. Neural networks are computational models inspired by the brain, consisting of interconnected neurons that learn patterns from data. Fig.5 demonstrates how a typical Neural Network works. Just like any other ML algorithm, neural network models have convincing accuracy in the field of rocket propulsion specifically, when predicting liquid engine wall temperatures and designing regenerative cooling systems [23].

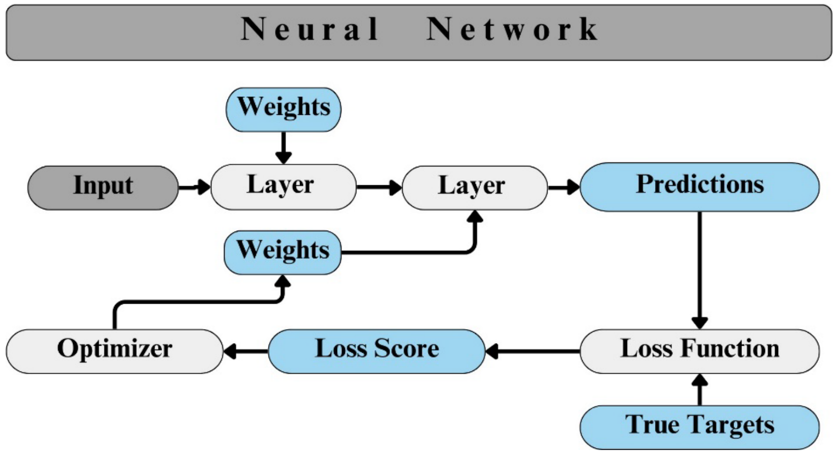


Figure 5. A flowchart depicting how a neural network works.

Fig.6 depicts what a typical neural network looks like for predicting wall temperatures in liquid rocket engines [4].

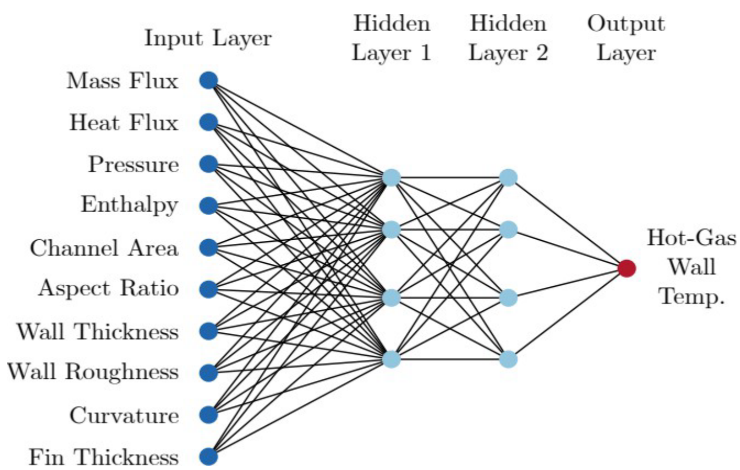


Figure 6. Example of a Neural Network architecture [4].

Neural Networks can also be used to predict fatigue life estimation of a liquid rocket engine [4]. The results (presented in results section) are convincing when compared to a traditional CFD simulation. A method for fault detection has been studied wherein two algorithms are combined namely Adaptive genetic algorithm (AGA) and Back propagation algorithm (BP). Genetic algorithm is an adaptive probability optimization technology based on biological genetics and evolutionary mechanisms and is suitable for optimizing complex systems [39]. BP is a multi-layer feed forward network trained by an error back-propagation algorithm. The topology of the BP neural network consists of an input layer, hidden layer, and output layer [39]. BP has a disadvantage of slow convergence speed; this is why a GA is used to optimize these disadvantages [39].

This fault detection follows steps in order to detect any malfunctions in the engine while running. The steps are as follows:

- Sensor data such as pressure, temperature, fuel flow rate, etc. are collected.
- This data is pre-processed.
- Initial engine sensor data us used to train the AGA-BP neural network model created to detect faults.

• Residual data is obtained by comparing the predicted sensor data values to the currently collected values.

• Compare this value with the set threshold, if residual value is greater than threshold value then engine malfunctions, if residual value is smaller than threshold value then engine is normal.

By using a BP neural network to forecast sensor data and then determining whether a fault occurs via a threshold detection mechanism, the technology can serve as an effective early warning system for engine failure.

Physics Informed Neural Network

Neural networks that are trained to solve supervised learning tasks while respecting any given law of physics described by general nonlinear partial differential equations can be regarded as Physics informed neural network [34]. PINN represents a novel class of neural networks that encode model equations, such as partial differential equations (PDEs), providing prior knowledge not found in traditional neural network models [24], this makes it a reliable candidate to predict fluid flows in nozzles during combustion, also making them ideal to be used to serve as an effective tool in improving existing combustors [30]. In this section, we would compare Physics informed neural network (PINN) with neural networks and review the systems in which PINN's can be used.

Combustion Modelling: PINN can be an excellent tool to model combustion as it involves complex chemical reactions and fluid dynamics. PINN's can extrapolate and can provide reliable predictions whereas Neural networks cannot, this is also one more advantage PINN's have over neural networks.

Table 1. Comparing between Physics Informed Neural Network and Artificial Neural Network.

Feature	Physics Informed Neural Network	Neural Network.
Use	It is used to solved complex physical systems with the help physical laws such as PDE's [33].	It is used for pattern recognition, approximation etc.
Data	It is less dependent on data as it uses the known physics to tackle complex systems.	It is more dependent on data.
Complexity	It is typically more complex due to the advanced level of calculus embedded while tackling a problem	It is generally simpler as it focuses more on the data provided to solve a particular system.
Domain Knowledge	It requires a significant amount of domain knowledge in order to define physical laws and their boundaries.	It requires less domain knowledge as it is dependent more on data provided.
Application	Fluid dynamics [29], heat transfer etc.	Image recognition, natural language processing etc.

Simulations: PINNs as mentioned above can also be used to understand propellant flows through injectors, valves and nozzles. Static tests can be expensive. PINNs can be used to minimize these tests (reducing expenditure) to improve combustion efficiency, fuel/ oxidizer ratio and optimize nozzle designs. All the examples are related to fluid dynamics and involve solving Partial differential equations (PDE's).

Heat Transfer: One of the many complexities involved in designing and manufacturing liquid propellant nozzles is managing the heat evolved through combustion. Managing heat is important as to avoid sudden heat fluxes in combustion chamber which would lead to combustion instabilities. Effective cooling systems are therefore necessary to maintain the optimum operating temperature, and this is where PINNs can be useful.

Maintenance: Failures and degradation of systems after multiple cycles is unavoidable. Predicting failures and maintaining rocket engines systems is very important as to ensure safety. PINNs can be used to model these processes as to understand at what cycle a thorough maintenance is required.

ORCA and GritBot

ORCA (Outlier-based Real-time Corrective Analysis): It defines an anomaly to be a point whose nearest neighbors in feature space are far away from it. It uses a novel pruning rule to obtain near-linear-time performance, allowing it to scale to very large datasets [11,32]. ORCA is mainly used to detect faults, detect outliers [25] and keep a real time monitoring on the sensor data which is being continuously received from rocket engine. One example is, Schwabacher has used to ORCA to look for anomalies in data from two rocket propulsion systems, the Space Shuttle Main Engine and rocket engine test stand E-1 at NASA Stennis Space Center [35].

GritBot: Rather than just looking for points that are anomalous with respect to the entire dataset, GritBot searches for subsets of the dataset in which an anomaly is apparent [11]. GritBot is thought of as a mature algorithm [31] which is used to detect Anomalies. GritBot can be used to recognize complex patterns in provided data.

Parameters	ORCA	GritBot
Sensor Data	Identifies Anomalies across various tests	Easier to define rules based on sensor types
Statistics	Identifies Outliers	Works with rules to improve analysis
Domain Knowledge	Requires fewer predefined rules	Effective for specific issues

Figure 7. Compares ORCA and GritBot.

Analysis of Methods and Discussion

This section discusses how each of the algorithm behaves and is used in different parts of the propulsion system to maximize its output. Fig.8 gives a comparative analysis of all the methods discussed in the second section of this review paper. All the parameters namely Anomaly detection, combustion optimization, sensor data analysis, inflight propellant mixture ratio and heat management presented in Fig.8 has previously been discussed in many papers [1, 4, 7, 9, 10, 11, 13, 23, 33, 39]. A, B, C, and D in Fig.8 depicts the favorability of the usage of the ML algorithms for that specific parameter. A being the most favorable and D being the least.

Learning Approach	Anomaly Detection	Combustion Optimization	Sensor Data Analysis	Inflight Propellant (O/F) Mixture Ratio Control	Heat Management
Unsupervised Learning	**B** (Can detect without labelled data)	**D**	**B** (Identifies trends in the given data	**D**	**D**

Supervised Learning	**D** (Need labelled data to detect anomalies)	**B** (Labelled Data needed)	**D**	**B** (Can predict if conditions are known)	**B** (Data should be provided for various situations encountered)
Neural Network	**A** (Highly effective)	**C**	**A** (Can derive complex relations between data)	**C**	C
Reinforcement Learning	**C** (Suitable simulation environment needed)	**A** (Suitable simulation environment needed)	**C**	**A** (Can derive optimal ration during flight)	**A** (Suitable simulation environment needed)

Figure 8. Summary of four most prominent ML algorithms to detect anomalies in LRP.

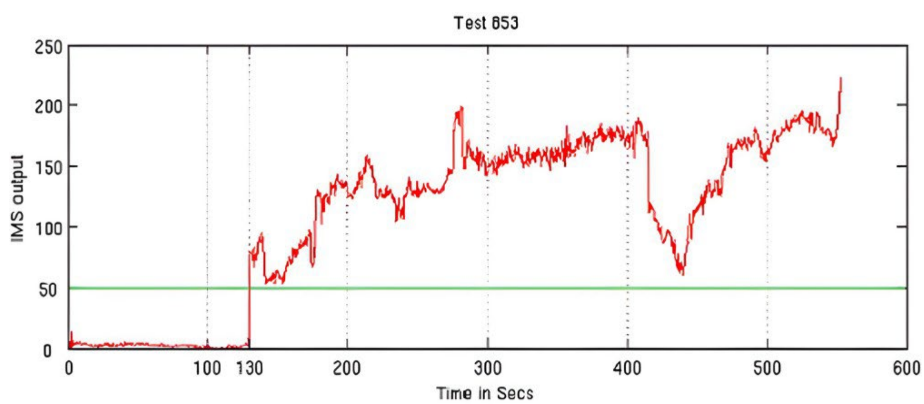


Figure 9. Failure detected by IMS at around 130 seconds into the test 853 [9].

Schwabacher et al. in their research [9] have used the historical data from the Space shuttle main engine (SSME) to present 9 anomalies detected by 4 unsupervised anomaly detection algorithms: ORCA, GritBot, IMS (Inductive monitoring system), SVM (Support vector machine) In the SSME data used, there are 90 sensors which measure temp, pressure, vibration, fuel flow rate and rotational velocity [9]. In the tests that they conducted, the 4 above mentioned algorithms were able to detect 1 major system failure and several sensor failures. They considered 4 anomalies in particular, first, the High-pressure fuel turbo pump (HPFTP) failure in which a turbine blade malfunctioned during SSME static test 853 approximately 130 s into the test (Fig.9) [9]. Fig.9 shows how an IMS algorithm detected a HPFTP failure at 130 seconds, the sudden spike in the data depicts anomalous behavior. A threshold of an IMS score of 50 (green line) is used to signal an alarm. Observed spikes in the data further may be due to some other anomalies present in the data.

The green line (y=50) represents a candidate threshold above which values are considered to be anomalous [9].

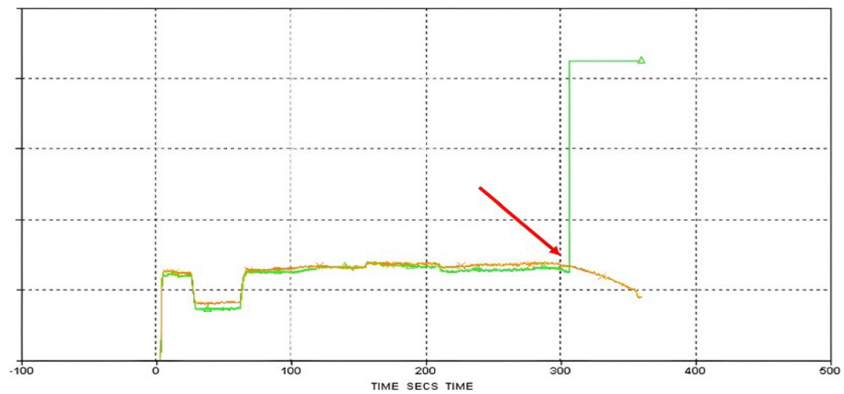


Figure 10. A temperature sensor failure at around 300 seconds was detected by GritBot [9].

The next anomalies were particularly detected by ORCA as well as GritBot. Fig.10 depicts values from 2 redundant temperature sensors. Both sensors present same values until 300 secs, but just then after, one of the sensors undergoes a failure, this particular failure was detected by GritBot. Similar to that, ORCA also detected a temperature sensor failure in a different engine test Fig.11. This failure, however, was detected not by comparing two redundant sensors during the same test, but by comparing two tests of two different engines [9].

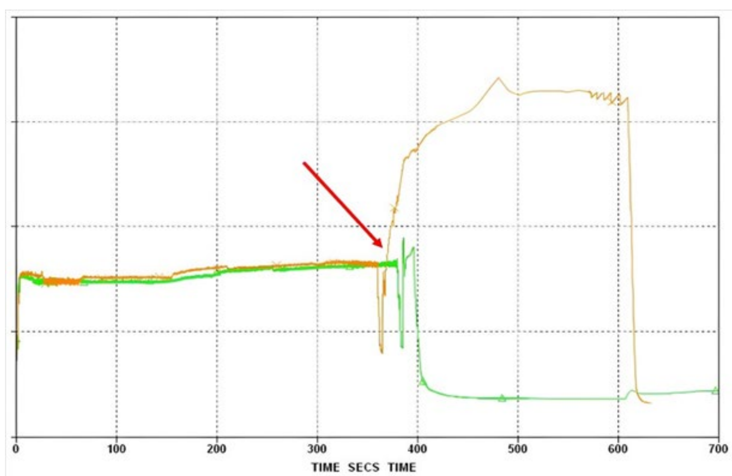


Figure 11. Like the anomaly detected by GritBot, ORCA also detected a similar temperature sensor failure at 360 seconds for a different test for a different engine [9].

ORCA and GritBot also detected 3 other failures, one of which was a pressure sensor failure, second was a vibration sensor failure and third was a SSME fuel flow anomaly [9]. A summary comparing ML algorithms and anomalies is provided by Schwabacher et al. in their paper. [9] Fig.12.

Anomaly	Orca	GritBot	IMS	SVM
HPFTP failure	Detected	Detected	Detected	Detected
Temperature sensor failure #1		Detected		
Temperature sensor failure #2	Detected			
Pressure sensor failure	Detected			
Vibration sensor failure	Detected			

Figure 12. A brief overview of which algorithm detected which failure [9].

Waxenegger-wilfing et al. have used a reinforcement learning (RL) algorithm technique to control liquid rocket engines [37]. As mentioned earlier if RL algorithms are given a suitable environment to train in, they can adapt and control the complexities involved in the liquid rocket engines. Waxenegger-wilfing et al. have used 2 traditional controllers namely Open loop system (OLS) and PID (Proportional, Integral and Derivative) to compare the performance with RL algorithms. The suitable environment we were talking earlier was given to the RL algorithm through EcosimPro (EcosimPro is a modeling and simulation tool for 0D or 1D multidisciplinary continuous and discrete systems [37]).

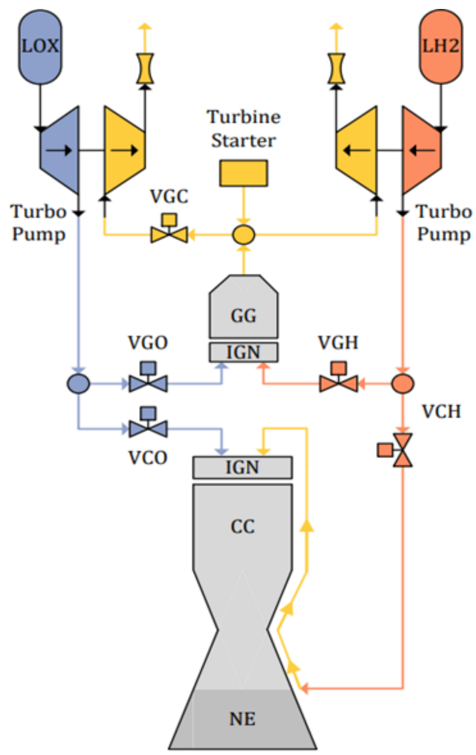


Figure 13. Engine architecture that was used to study the suitability of an RL approach [37].

Fig.13 provides the engine architecture that was used to study the suitability of an RL approach. 3 main variables were considered when comparison was done between OLS, PID and RL. The variables were mixture ratio in gas generator (MRGG), mixture ratio in pump inlet (MRPI) and combustion chamber pressure (PCC). Three algorithms mentioned above are used to control all the three variables and then compared. The OLS works well for nominal conditions but leads to significant deviations from desired values [37]. PID controllers achieved somewhat better performance than open loop control system. RL algorithms performed the best giving the least deviations from desired values. Fig.14 summarizes the data observed over several tests conducted by Waxenegger-wilfing et al.

Target	Algo.	Reward	Steady-State Values			*IAE*		
p_{CC} (bar)		cum. (–)	p_{CC} (bar)	MR_{GG} (–)	MR_{PI} (–)	p_{CC} (bar)	MR_{GG} (–)	MR_{PI} (–)
100	OLS	-7.9	100.0	0.90	5.18	591	4.7	27
	PID	-6.5	98.9	0.90	5.17	632	4.0	19
	RL	-4.2	99.9	0.90	5.18	519	2.8	13
80	OLS	-7.0	79.7	0.90	5.20	576	6.0	15
	PID	-5.2	79.9	0.90	5.20	433	3.8	9
	RL	-4.4	80.8	0.90	5.18	366	2.9	8

Figure 14. Controller performance for nominal turbine efficiencies [37].

Controller	Performance	Pressure Control	Mixture Ratio Control
RL	Best	Precise	Closest to Target
PID	Good	Closer to Target	Some Deviations
OLS	Worst	Acceptable	Large Deviations

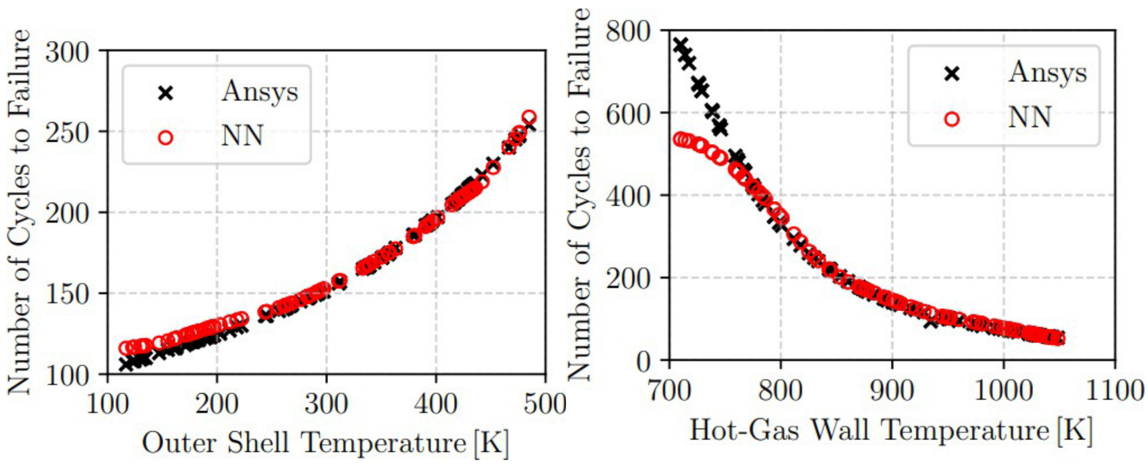
Figure 15. A simplified version of the figure 14.

Fig.15 simplifies the data presented in Fig.14. The target value 100/ 80 sets a reference target value for these controllers. The lower the reward value the better meaning that it has deviated far less from the reference value, which in this case is 100 or 80. The IAE meaning Integral absolute value is a metric that quantifies the total deviation of a controlled variable (pressure or mixture ratios) from its desired value throughout the simulation. As stated earlier, in this case as well, lower the value better it is as the ideal observations from the simulations should have small deviations. Overall, the RL algorithm as seen from the data outperforms both OLS as well as PID controllers. The highest deviation observed from the 100-bar target simulation is from OLS which is -7.9, whereas RL delivers the lowest deviation -4.2. similarly, with the 80-bar target simulation OLS performs the worst compared with the 3 and RL performs the best. Therefore, this data suggests that OLS is not the ideal choice for maintaining optimal combustion conditions is rocket engines whereas RL can. The RL controller's adaptability in real-time allows it to maintain optimum pressure and mixture ratios, leading to a greater efficiency while combustion.

Simulations and models are so important in any field, when discussing about rocket they are even more com- plex and requires tedious amounts of calculations. Though these calculations can be done by high fidelity computational fluid dynamics (CFD) or finite element method (FEM), these calculations take humongous efforts. Neural Networks (NN) on the other hand can build surrogate models which can produce near similar results compared to FEM or CFD. These NN surrogate models can reduce the efforts for calculations and make task easier.

These NN models can be used to predict regenerative cooling of liquid rocket engines. Waxenegger-wilfing et al. have already built a NN based surrogate model for the maximum wall temperature along the cooling channel [4]. The NN employs a feed forward architecture with 4 hidden layers and 408 neurons per layer, the authors have trained the NN model using data from approximately 20000 CFD simulations [4]. Fig.16 displays exemplary NN architecture. The NN model was used to study the cooling channel performance of the LUMEN engine.

Waxenegger-wilfing et al. have also developed a NN based surrogate model for fatigue life estimation [4]. Fatigue life estimation is one of the most important parts to remember when designing any reusable liquid rocket engine. Specifically, very essential when designing nozzles and thrust chambers for rockets. The NN's in the model are trained by Dresia et al. using samples of the computationally expensive calculations [4]. Approximately 120,000 data points were used for training which would have helped to make the NN architecture more robust. As observed, the NN model has achieved near perfect simulation when compared to Ansys. Overall, the model estimates the number of cycles to failure with a mean squared error (MSE) of 239 on previously unseen data (equal to a mean percentage error of 7) [4]. The results show that NN's are not able to extrapolate, which is what we discussed in the methodologies section. To summarize NN's performance have been up to the mark and can be thought of as a tool we can use in future.



a) Predictive performance of Neural network for the outer shell temperature per cycle to failure.

b) Predictive Performance of Neural network for the hot gas wall temperature per cycle to failure.

Figure 16. Performance of the Neural Network for the combustion chamber fatigue life prediction [4].

Conclusion

In this review paper we discuss about the various Machine learning algorithms used in the field of liquid rocket propulsion systems. Few of the most prominent algorithms such as Supervised, Semi supervised, Unsupervised, Reinforcement, neural networks and Physics informed neural networks are discussed in this review paper which focus on improving combustion efficiency, propellant flow rate, minimizing inflight instabilities, etc. The methodology introduces different types of algorithms, following that, the results section gives you a practical viewpoint of where these algorithms fail and succeed, every algorithm has its advantages as well as disadvantages. As seen, many of these algorithms are heavily dependent on sensor data. Pattern recognition and identification is what most of these algorithms perform internally. It is due to these ML algorithms, designing and optimization of liquid rocket engines have become more efficient and robust.

Limitations and Future Work

Detecting anomalies today can be done by various ML algorithms such as supervised/ unsupervised learning. These methods heavily depend on data which is being fed into such processes to analyze anomalies. In the future, dependency on data can be reduced by focusing more on algorithms such as Physics informed neural networks which uses its own knowledge. Physics informed neural networks particularly can be very useful in predicting propellant flows, combustion, heat fluxes etc. as it incorporates partial differential equations (PDE's) to govern any physical system. As we increase the usage of ML in designing, optimizing and operating liquid propulsion systems for rockets in the future, it is very necessary to ensure safety. Real time diagnosing is one more aspect we should consider in future research.

RL algorithms can be used to control thrust during soft (retro) landing a rocket. If given an ideal environment to train in, RL may enable engine control systems [37]. Regarding the use of reinforcement learning (RL) to control liquid rocket engines in the future, it's important to note a few things: RL approach cannot guarantee specific stability within the control system, and implementing hardware constraints may pose additional challenges [37].

One problem every ML modelling process faces is the uncertainties in the data as well as the presented predictions. In the future, the Gaussian processes can be used to create surrogate models of complex propulsion systems. Gaussian processes not only predict but also provide a measure of uncertainty in those predictions. For the matter of fact, amazing successes have already been achieved by using these algorithms [34]. Bayesian approximation can also be used to quantify these uncertainties to accept ML methods in safety applications [34].

There are many variables that need to be considered when designing a particular neural network or a particular ML algorithm in order to get accurate results. Besides variables like chamber pressure, temperature, propellant flow rate etc. one variable that we miss is time. In the future, algorithms which model time can be explored, one such method is windowing, for each time step the algorithm considers the sensor values at that time step and at a fixed number of previous time steps in determining whether the point is an anomaly [23].

Machine learning algorithms are enhancing efficiency of liquid rocket propulsion systems. In the future by incorporating ML methods, human space missions to moon, mars can ensure safety as well as reliability.

Acknowledgment

I am very thankful to my mentors namely Dr. Raj Dandekar, Dr. Rajat Dandekar and Dr. Sreedath Panat for guiding me and teaching me the fundamentals of Machine Learning.

References

[1] Jianjun Wu. Liquid-propellant rocket engines health-monitoring—a survey. *Acta Astronautica*, 56(3):347–356, 2005.

[2] Zhigang Feng and Qi Wang. Research on health evaluation system of liquid-propellant rocket engine ground-testing bed based on fuzzy theory. *Acta Astronautica*, 61(10):840–853, 2007.

[3] Richard Strunz and Jeffrey W Herrmann. Reliability as an independent variable applied to liquid rocket engine hot fire test plans. *Journal of propulsion and power*, 27(5):1032–1044, 2011.

[4] Günther Waxenegger-Wilfing, Kai Dresia, Jan Deeken, and Michael Oschwald. Machine learning methods for the design and operation of liquid rocket engines–research activities at the dlr institute of space propulsion. *arXiv preprint arXiv:2102.07109*, 2021.

[5] Hao Li and Cor-Paul Bezemer. Studying popular open source machine learning libraries and their cross-ecosystem bindings. *arXiv preprint arXiv:2201.07201*, 2022.

[6] Dheeraj Yadav. Machine learning: Trends, perspective, and prospects. *Science*, 349:255–260, 2020.

[7] Nobendu Sen, Gautham M Nair, and A Immanuel Selvakumart. Machine learning in combustion:

Optimization of fuel combustion of rockets. *Acceleron Aerospace Journal*, 2(4):241–250, 2024.
[8] Steven L Brunton, J Nathan Kutz, Krithika Manohar, Aleksandr Y Aravkin, Kristi Morgansen, Jennifer Klemisch, Nicholas Goebel, James Buttrick, Jeffrey Poskin, Adriana W Blom-Schieber, et al. Data-driven aerospace engineering: reframing the industry with machine learning. *AIAA Journal*, 59(8):2820–2847, 2021.
[9] Mark Schwabacher, Nikunj Oza, and Bryan Matthews. Unsupervised anomaly detection for liquid-fueled rocket propulsion health monitoring. *Journal of aerospace computing, information, and communication*, 6(7):464–482, 2009.
[10] Rodney A Martin. Unsupervised anomaly detection and diagnosis for liquid rocket engine propulsion. In *2007 IEEE Aerospace Conference*, pages 1–15. IEEE, 2007.
[11] Mark Schwabacher. Machine learning for rocket propulsion health monitoring. *SAE transactions*, pages 1192 1197, 2005.
[12] Oskar J Haidn. Advanced rocket engines. *Advances on propulsion technology for high-speed aircraft*, 1:6–1, 2008.
[13] Kai Dresia, Günther Waxenegger-Wilfing, Robson Henrique Dos Santos Hahn, Jan C Deeken, and Michael Oschwald. Nonlinear control of an expander-bleed rocket engine using reinforcement learning. 2021.
[14] Daniel Klinger, Alex Casey, Tim Manship, Steven Son, and Alejandro Strachan. Prediction of solid propel- lant burning rate characteristics using machine learning techniques. *Propellants, Explosives, Pyrotechnics*, 48(4):e202200267, 2023.
[15] Vladimir Nasteski. An overview of the supervised machine learning methods. *Horizons. b*, 4(51-62):56, 2017.
[16] Nur Amalina Diyana Suhaimi and Hafiza Abas. A systematic literature review on supervised machine learning algorithms. *Perintis Ejournal*, 10(1):1–24, 2020.
[17] Batta Mahesh. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, 9(1):381–386, 2020.
[18] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.
[19] Paul R Gradl and Christopher S Protz. Channel wall nozzle manufacturing technology advancements for liquid rocket engines. In *International Astronautical Congress (IAC), 2019*, number M19-7683, 2019.
[20] Hao Ma, Yu-xuan Zhang, Oskar J Haidn, Nils Thuerey, and Xiang-yu Hu. Supervised learning mixing characteris- tics of film cooling in a rocket combustor using convolutional neural networks. *Acta Astronautica*, 175:11–18, 2020.
[21] John A Halchak, James L Cannon, and Corey Brown. Materials for liquid propulsion systems. Technical report, American Institute of Aeronautics and Astronautics, 2018.
[22] R Amri and T Rezoug. Numerical study of liquid propellants combustion for space applications. *Acta Astronautica*, 69(7-8):485–498, 2011.
[23] Kai Dresia, Eldin Kurudzija, Jan Deeken, and Günther Waxenegger-Wilfing. Improved wall temperature prediction for the lumen rocket combustion chamber with neural networks. *Aerospace*, 10(5):450, 2023.
[24] Mingming Guo, Xue Deng, Yue Ma, Ye Tian, Jialing Le, and Hua Zhang. Hypersonic inlet flow field reconstruction dominated by shock wave and boundary layer based on small sample physics-informed neural networks. *Aerospace Science and Technology*, 150:109205, 2024.
[25] David L Iverson, Rodney Martin, Mark Schwabacher, Lilly Spirkovska, William Taylor, Ryan Mackey, J Patrick Castle, and Vijayakumar Baskaran. General purpose data-driven monitoring for space operations. *Journal of Aerospace Computing, Information, and Communication*, 9(2):26–44, 2012.
[26] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.
[27] Tobias Kaiser. *Optimal control of liquid propellant rocket engines for landing of reusable stages*

using deep reinforcement learning. PhD thesis, Master's thesis, Universitat Wurzburg, 2021.
[28] Martin Q Ma, Yue Zhao, Xiaorong Zhang, and Leman Akoglu. The need for unsupervised outlier model selection: A review and evaluation of internal evaluation strategies. *ACM SIGKDD Explorations Newsletter*, 25(1):19–35, 2023
[29] Zhiping Mao, Ameya D Jagtap, and George Em Karniadakis. Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering*, 360:112789, 2020.
[30] Sathesh Mariappan, Kamaljyoti Nath, and George Em Karniadakis. Learning thermoacoustic interactions in combustors using a physics-informed neural network. *arXiv preprint arXiv:2401.00061*, 2023.
[31] Rodney A Martin. An investigation of state-space model fidelity for ssme data. In *2008 International Conference on Prognostics and Health Management*, pages 1–12. IEEE, 2008.
[32] BRYAN MATTHEWS and ASHOK N SRIVASTAVA. Comparative analysis of data-driven anomaly detection methods on solid rocket motor faults.
[33] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
[34] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.
[35] Mark Schwabacher. A survey of data-driven prognostics. *Infotech@ Aerospace*, page 7002, 2005.
[36] Wei Shyy, P Kevin Tucker, and Rajkumar Vaidyanathan. Response surface and neural network techniques for rocket engine injector optimization. *Journal of Propulsion and Power*, 17(2):391–401, 2001.
[37] Günther Waxenegger-Wilfing, Kai Dresia, Jan Deeken, and Michael Oschwald. A reinforcement learning approach for transient control of liquid rocket engines. *IEEE Transactions on Aerospace and Electronic Systems*, 57(5):2938–2952, 2021.
[38] Günther Waxenegger-Wilfing, Ushnish Sengupta, Jan Martin, Wolfgang Armbruster, Justin Hardi, Matthew Juniper, and Michael Oschwald. Early detection of thermoacoustic instabilities in a cryogenic rocket thrust chamber using combustion noise features and machine learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(6), 2021.
[39] Huahuang Yu and Tao Wang. A method for real-time fault detection of liquid rocket engine based on adaptive genetic algorithm optimizing back propagation neural network. *Sensors*, 21(15):5026, 2021.
[40] Lodewijk Kallenberg. Markov decision processes. Lecture Notes. University of Leiden, 428, 2011.