# ABSTRACT

In recent years, Deep Learning has promoted the rapid development of artificial intelligence and penetrated into various fields, but the massive data transmission of Artificial Intelligence (AI) training in network will affect the real-time performance of the system. As is known to all, real-time performance is very important in industrial field, especially in motion control. If the system is responding slowly, it cannot receive or send data acquisition and control commands on time. Finally, this will hinder the popularization of AI technology in the field of industrial automation. In this paper, we present a Three-Real-Time architecture, it involves real-time hardware network, real-time operating system and real-time scheduling virtualization layer. We hope to conceive a new industrial edge computing model after instead a traditional industrial controllers. Thus, the application of artificial intelligence technology in industrial automation can be thoroughly solved.

Keywords: three-real-time; artificial intelligence; embedded-AI; edge computing; industrial automation

# TABLE OF CONTENTS

1. INTRODUCTION
2. LITERATURE REVIEW
3. METHODOLOGY
4. RESULTS & COMPARISION

5 OUTPUTS

6 CONCLUSION

7 REFERENCES

8 SAMPLE CODE

9 APPENDIX

## LIST OF CONTENTS

ABSTRACT

TABLE OF CONTENTS

LIST OF FIGURES

### INTRODUCTION 10

### LITERATURE REVIEW 24

* 1. The challenge of the AI field 26
  2. An overview of the AI field 27
     1. Reasoning 27
  3. Expert system 28
  4. Natural language understanding 28

### METHODOLOGY 30

* 1. Industrial automation using RPA 31

3.2 RPA Lifecycle 32

3.2.1 Analysis Phase 33

* + 1. Design Phase 33

3.2.3 Deployment Phase 33

3.2.4 Control and Monitoring Phase 33

**4 RESULTS AND COMPARISIONS 39**

4.1 RPA and other Enterprise Automated Tools 40

* 1. RPA works 41

**5 OUTPUTS** **44**

**6 CONCLUSION** **50**

**7 REFERENCES 53**

**8 SAMPLE CODE 55**

**APPENDIX**

**LIST OF FIGURES**

**Fig. No. Description Page No.**

1.1 Block diagram of Machine Learning 4

1.2 Working of Supervised Learning 5

2.1 Illustration Concerning diverse fields of AI 17

3.1 Three step Framework for Data preparation 24

3.2 Working of Random Forest Algorithm 26

3.3 IOT based model 29

4.1 Content Distribution Network 41

5.1 Screenshot on selecting sensor temperature 43

5.2 Screenshot on selecting temperature 25oC 43

5.3 Screenshot on selecting temperature 30oC 44

5.4 Screenshot on selecting sensor pressure 44

5.5 Screenshot on selecting pressure 9Pascals 45

5.6 Screenshot on selecting pressure 15pascals 45

5.7 Screenshot on selecting sensor gas 46

5.8 Screenshot on selecting gas 150ppm 46

5.9 Screenshot on selecting gas 200ppm 47

### 

**CHAPTER – 1**

**INTRODUCTION**

1. **INTRODUCTION**

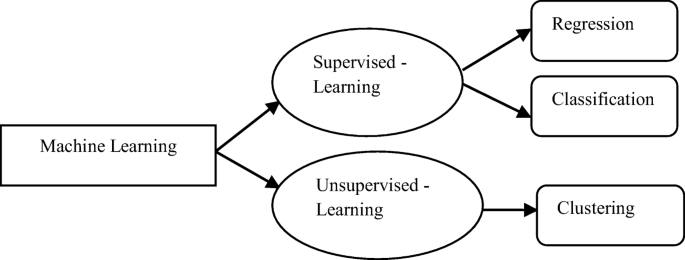
Maximum number of industries and organizations depends on the usage of machines, electronic widgets, and various recourses for completing different tasks. All are operated with the help of machines. Even though there are so many benefits to us, running with them without any safety aspects will cause injury and even leads to loss of life (Joseph Zulick [2019](https://link.springer.com/article/10.1007/s42797-020-00020-y#ref-CR8); Fairoze et al., [2018](https://link.springer.com/article/10.1007/s42797-020-00020-y#ref-CR9)). Machines can works on relevant data or inputs (Bradley et al., [1991](https://link.springer.com/article/10.1007/s42797-020-00020-y#ref-CR12)). But preparing relevant data to solve a particular problem is a challenging task to engineers. And make sure that the relevant data is in a useful scale and correct format with meaningful features should include.

Novelty and adaptation are tremendously significant to the industrial automation (Raffaele Cioffi et al., [2020](https://link.springer.com/article/10.1007/s42797-020-00020-y#ref-CR13)). As technology advances while the development of standards, designers are often left with future predictions. This causes them to overestimate or underestimate the necessary safety functions (Tina Hull [2019](https://link.springer.com/article/10.1007/s42797-020-00020-y#ref-CR4)). Safety system helps us take some crucial decisions in Industries. Usage of Internet of Things the system becomes secured and live data monitoring is also possible (Vijayalakshmi and Muruganand, [2017](https://link.springer.com/article/10.1007/s42797-020-00020-y#ref-CR2); Deshpande and Sangitasanap, [2016](https://link.springer.com/article/10.1007/s42797-020-00020-y#ref-CR10)). Large variety of industrial Internet of Things and its applications have been prepared and used in recent years (Da Xu et al., [2014](https://link.springer.com/article/10.1007/s42797-020-00020-y#ref-CR3); Khan and Bhat, [2014](https://link.springer.com/article/10.1007/s42797-020-00020-y#ref-CR5)).

Machine Learning (ML) is one of the emergent platforms for automation, through which new advancements are made and easily monitor as well control the system using datasets. ML is a part of computer science that often uses arithmetical techniques to give computers the ability to be trained with datasets, without programming (S. Kavitha et al., [2019](https://link.springer.com/article/10.1007/s42797-020-00020-y#ref-CR1)). Machine Learning, Artificial Intelligence and Internet of Things with software engineering practices are very useful practices in providing safety in industries.

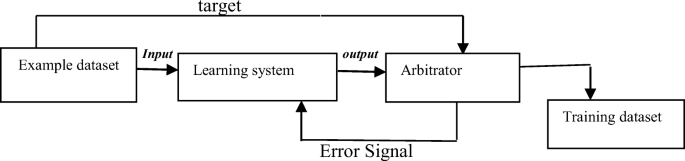
Machine Learning is a notion which allows the system to learn from experiments. So that it can recognize patterns make predictions and complete tasks (Gatis Mikelsons et al., [2019](https://link.springer.com/article/10.1007/s42797-020-00020-y#ref-CR11)). So instead of writing the code, just feed data by training it repeatedly with datasets. Machine learning involves training machines with some existing data sets or past data and produce algorithms that the machine can learn from without any explicit programming and help with the predictive analysis of the future data sets.

Basically there are training, validation and testing data sets. We train the classifier using training set, validation set adjust the parameters and then test set, test the performance of classifier. Machine learning implementations shown in fig. [1](https://link.springer.com/article/10.1007/s42797-020-00020-y#Fig1) are classified into three major categories (supervised, unsupervised and semi-supervised learning) (Aurangzeb Khan et al., [2010](https://link.springer.com/article/10.1007/s42797-020-00020-y#ref-CR6); Cabe Atwell [2017](https://link.springer.com/article/10.1007/s42797-020-00020-y#ref-CR7)) depending on the scenery of the knowledge ‘response’ or ‘signal’ accessible to a learning system. Here Supervised Learning learns from example dataset and related target responses that can consist of numeric values or string labels in order to later predict the correct response when posed with new examples.

[](https://link.springer.com/article/10.1007/s42797-020-00020-y/figures/1)

**Fig. 1.1 Block diagram of Machine Learning (ML)**

Figure 1.1 depicts the working of supervised learning. Another categorization of machine learning tasks (Classification, Regression, and Clustering) takes place when one believes the preferred output of a machine learned system. And proposed methodology uses classification technique.

[](https://link.springer.com/article/10.1007/s42797-020-00020-y/figures/2)

**Fig. 1.2** **working of supervised learning**

After an extended period out of the limelight,\* artificial intelligence, or AI, has returned to the public consciousness in a big way. AI’s virtues and vices are now discussed daily in the popular press. While the societal implications of AI remain a topic of debate, it is broadly accepted that its business implications will be significant. Among those who track such trends, AI is expected to be a large driver of enterprise competitiveness in the not-so-distant future.

The views shared in a recent report by investment bank Goldman Sachs are representative of this sentiment. The paper states that “the ability to leverage AI technologies will become one of the major defining attributes of competitive advantage across all major industries in the coming years. While the strategy will differ by company size and industry, management teams that don’t focus on leading in AI and benefiting from the resulting product innovation, labor efficiencies, and capital leverage risk being left behind.”

1 IT industry research firm Gartner anticipates this impact as well, and projects it to take root sooner than later. They foresee that “by 2018, more than half of large organizations around the globe will compete using advanced analytics and proprietary algorithms, causing disruption on a grand scale.”

2 The question for enterprises then is not when and if, but how.

Unfortunately, answering even this basic question can be difficult today, because doing so requires a broad and clear understanding of the various ways that AI can impact the business. While much information on the topic exists, it tends to be very fragmented, poorly organized, and limited to only a small subset of use cases. For better or worse, contemporary discussion of enterprise AI use cases has focused on applications in the digital domain. These are applications like getting people to click on ads, making recommendations, personalizing the customer experience, predicting customer churn, and detecting fraud of various sorts. But what about those parts of an organization whose operations extend beyond the digital domain? Surely they need AI too?

The answer, we believe, is a resounding yes. In fact, AI presents a unique and compelling opportunity for those businesses whose operations span the virtual and physical worlds. To compete effectively, these firms must drive towards increased operational efficiency and asset utilization, and they must aspire to the same level of agility in the physical aspects of their business as they have sought in its virtual aspects. It has become clear that some of the world’s largest enterprises believe this as well. They are betting big on AI to create competitive advantage through increased situational awareness, greater efficiency, and higher quality. Consider a few examples:

• In the five years since Amazon’s $775 million purchase of Kiva to form Amazon Robotics, the company has invested heavily in AI techniques, including sponsoring an annual Robo Cup competition “to strengthen the ties between the industrial and academic robotic communities to promote shared and open solutions to some of the big problems in unstructured automation.”

• Supply chain services provider J.B. Hunt announced a five-year, $500 million effort to develop technology that will connect shippers and carriers by using real-time data and artificial intelligence to match freight with capacity.

• Boeing has acquired Liquid Robotics, developer of autonomous maritime systems, It has established a joint Analytics lab with Carnegie Mellon University, It has partnered with Microsoft, To use its Azure cloud platform to improve the operational efficiency of aircraft.

• German manufacturer Bosch announced a €300 million investment in the new Bosch Center for Artificial Intelligence. According to Dr. Volkmar Denner, chairman of the board of management of Bosch, “Ten years from now, scarcely any Bosch product will be conceivable without artificial intelligence. It will either possess that intelligence itself, or AI will have played a key role in its development or manufacture.”

Our hope is that this paper will help illuminate where and how enterprises can apply AI to drive greater operational agility and performance across the set of use cases we call “industrial AI.” We also discuss the challenges in and impediments to doing so, and offer some pointers for those organizations just getting started. We begin by defining AI and Industrial AI.

What is Industrial AI?

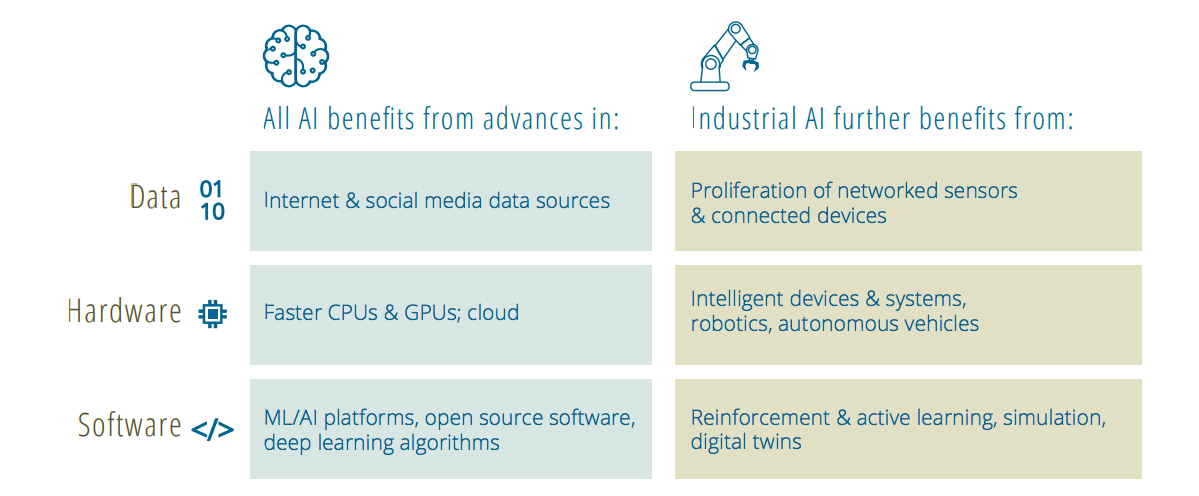
To define industrial AI, we must first define AI itself. Although the field of artificial intelligence has existed for over half a century, it has no clear and all-encompassing definition. Further, the lines between AI and adjacent fields like machine learning, big data, predictive analytics, and IoT are often blurred, as are the lines between AI and subfields like deep neural networks and cognitive computing.

For our purposes, artificial intelligence refers to those computer science techniques and technologies that allow software to exhibit ‘smarts’—in other words, to do things that seem human-like. This can include things like making decisions, recognizing objects, or understanding speech. It really is a very broad term. Strictly speaking, machine learning (ML) is a subset of AI. ML refers to a set of techniques that allow us to create AI software by training that software with data) to display some desired intelligent behavior.

This is as opposed to, for example, explicitly programming our software with a bunch of rules to generate our desired behavior—and it’s a very powerful concept. It is for this reason that, while machine learning is only one way to build an artificially intelligent system, for all practical purposes ML and AI are used interchangeably today.

All the interesting activity in AI is in machine learning.† What about cognitive computing? It’s a bit more esoteric a term, usually used to highlight capabilities akin to humans’ higher level thinking and reasoning skills. An example would be the ability to determine the sentiment expressed in text or images, or what objects are present in pictures. But again, for all practical purposes, the term is most often used interchangeably with AI—in fact, it’s the preferred term in some regions of the world—and the work in this field is based upon machine learning. How does this relate to big data? Well, data is used to train the machines, and the more you have of it the better (assuming it’s high quality data).9 And how about predictive analytics? Well, machine learning can be a more powerful way to make predictions, and one that can learn from patterns in the data. But simple averages and other formulas can be used for predictions as well… these need not be based on ML/AI.

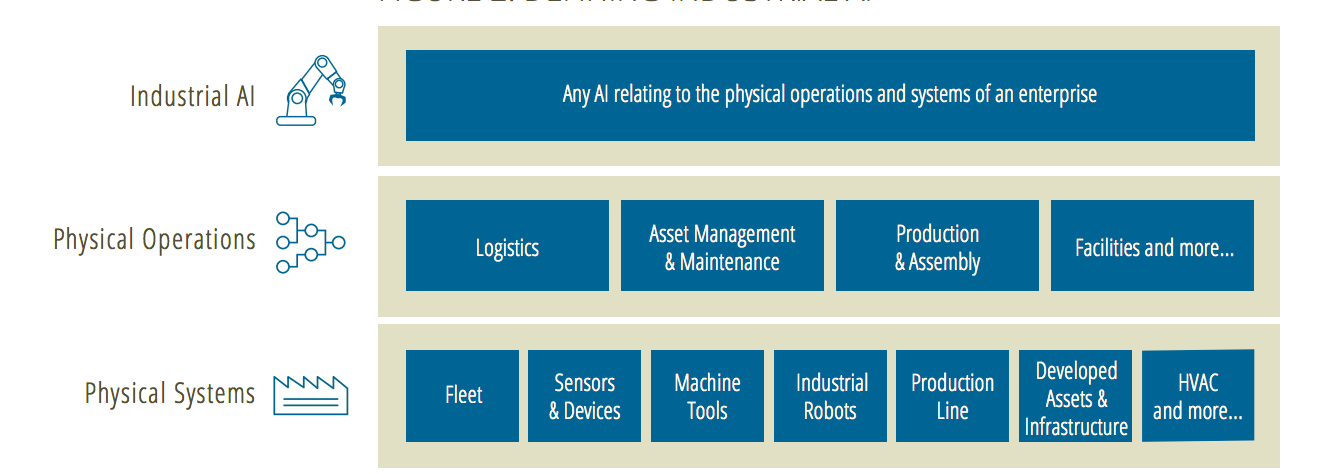
Finally, for enterprises whose operations involve the physical world, the industrial internet of things (IoT/IIoT) is an increasingly important source of insight into the status, location and performance of enterprise assets . Because IoT devices and sensors can number into the millions, and can report status with millisecond resolution, the resulting data volumes can quickly become voluminous, lending themselves to the application of machine learning techniques.



The growing ubiquity of data, high-performance hardware and sophisticated software have all contributed to laying the groundwork for the current resurgence in artificial intelligence. A similar set of trends has created a rich opportunity in industrial AI.

To what then does the term “industrial AI” refer? Well, certainly the word industrial has certain immediate connotations, primarily of manufacturing and heavy industry. But to limit our scope to just those industries would be to miss the less obvious connections between a broad set of related use cases, the environments they exist within, and the common challenges and requirements that they give rise to.

According to this definition, industrial AI includes, for example, applications relating to the manufacture of physical products, to supply chains and warehouses where physical items are stored and moved, to the operation of building HVAC systems, and much more. Any company in any industry can have opportunities to apply industrial AI.



Due to the physical nature of the systems and processes to which they relate, industrial AI systems share similar characteristics and constraints. For example, the fact that industrial AI ultimately relates to the physical systems of an enterprise tends to mean that access to training and test data is more difficult; the reliance on subject matter expertise is larger; the AI models themselves are harder to develop, train, and test; and the costs

their failure are greater. In other words, the stakes are higher. We elaborate on the significance of this in the next section.

How is Industrial AI Different?

A big part of what makes industrial AI different from consumer and other business applications of AI is the fact that the stakes are much higher. Exploring this notion further will help clarify these differences. Consider one example: the case of a predictive maintenance system monitoring performance of an aircraft engine. In a recent Forbes article, Harel Kodesh, former vice president and CTO of GE Software notes that “if an analytical system on a plane determines an engine is faulty, specialist technicians and engineers must be dispatched to remove and repair the faulty part. Simultaneously, a loaner engine must be provided so the airline can keep up flight operations. The entire deal can easily surpass $200,000.”10

Clearly the cost of a “false positive” here is greater than the cost of Netflix showing the wrong movie recommendation, or Amazon upselling the wrong product. But the differences go further. This system is likely subject to any number of compliance requirements, and the system’s recommended action might trigger a variety of reporting actions.

The development of the predictive model is likely significantly more involved than building a recommender: a variety of live and simulated engine sensor data must be captured; the sensor data likely requires extensive cleaning before use; the model must be trained against the cleansed data; and it must be tested against a test dataset, in simulation, and in production.

This process likely relies heavily on a variety of subject matter experts including systems engineers, maintenance and performance engineers, and more, not to mention the software engineering talent involved.

Data acquisition and storage.

Unlike “born digital” data captured, for example, from web interaction logs, industrial AI systems often rely on data captured from sensors that seek to represent the real world digitally. Unfortunately, this process can result in inherently noisy datasets. Sensor data can also be voluminous. Acquiring this data and storing it for analysis can be extremely complex. Furthermore, because of the cost of generating training data under a wide variety of conditions, simulation is often used. High-fidelity simulations, or “digital twins,” can be very effective, but can also be difficult to create and maintain, and computationally expensive to run.

**Training challenges**:

Much of the recent fanfare around AI has been based on the success of “deep learning.” In most cases, these successes are based on supervised learning style problems in which deep neural networks are trained with labeled training data. While it can be difficult in any domain to collect the volume of labeled training data required to effectively train machine learning models, this can be particularly challenging in industrial scenarios in which few examples of the most interesting “black swan” events—such as part or product failures—occur. This increases the complexity of training and thus the overall cost of developing the machine learning system.

**Testing costs and complexity**:

Testing AI systems on operating production lines, industrial equipment, warehouses and other industrial systems is both expensive and disruptive. Because of this, industrial AI systems are often trained and tested using simulation, the challenges of which have already been discussed.

**High regulatory requirements:**

Industrial environments are often subject to compliance statutes that impact operations, including technical, legal and corporate requirements, and governmental regulations. Depending on the market and industry, compliance requirements span areas such as product safety, public/employee health and safety, environmental impact, and workplace safety, but they can also directly specify controls around automation systems, as does, for example, the European Machinery Directive.11, Regulatory controls, which often require that changes to industrial processes be extensively validated and verified, can be at odds with the goals of automation via AI, which encourage rapid adaptation of processes via closed-loop feedback.13

**High cost of failure & change:**

As we saw in our aircraft engine example, it is common in industrial scenarios for the cost of failure to be extremely high. The cost of change is similarly high. When an enterprise has many millions of dollars invested in factories and warehouses, automation technology—AI or otherwise—must either work with those existing investments or demonstrate extremely compelling ROI.

**Large state spaces:**

Modern industrial systems are extremely complex, often offering tens or hundreds of inputs over which machine learning algorithms may optimize. This can make for more complex development and training routines (both in terms of time and cost) and can require the use of sophisticated techniques to simplify the problem and ensure convergence to a solution

**Cost of talent:**

Data scientists, data engineers and data-savvy programmers and subject matter experts are the backbone of the team required to implement AI solutions. These skills are both rare and expensive in today’s employment market, and firms must compete for top talent with internet leaders like Facebook and Google.

Why Industrial AI?

Given higher stakes, what’s to motivate enterprises to do anything more than the status quo? In other words, why should enterprises care about industrial AI? From a macroeconomic perspective, today’s enterprises, particularly those with physical-domain operations, are under tremendous competitive pressure.

They are competing not only with VC-funded internet companies with disruptive business models and the ability to operate at a loss for extended periods of time, but also with overseas competitors with fundamentally different economics and constraints.14 This has lead enterprises to aggressively seek out opportunities to reduce costs, increase efficiencies, and innovate in their business models, the latter often under the banner of “digital transformation.” AI is an important tool for achieving all these things, and, for these reasons, has long been of interest to enterprises with industrial operations. Industrial AI use cases abounded in the 80s and early 90s, prior to the most recent “AI winter.”15 During this period, large firms including Boeing, JPL, Intel, and others, invested heavily in expert systems and other forms of AI to drive greater efficiencies. In many ways, the industrial AI of the 80s and 90s and the industrial AI of today share common goals. With expert systems, enterprises of the former era sought to transfer knowledge from engineers and other subject matter experts (SMEs) to intelligent computer systems that could guide and assist humans in the performance of their work. Now, with both more sophisticated AI and more pervasive automation, we seek to transfer the knowledge to more intelligent, more powerful systems that can both assist humans and perform some of their tasks.

The specific benefits of such artificial intelligence systems in industrial environments are many. At a high level, they include: Enhanced, and predictive, situational awareness. By allowing enterprises to model complex industrial systems, industrial AI allows enterprises to increase quality, reduce downtime, avoid stock-outs, reduce risk, and more. Better planning and decision-making.

By helping enterprises assess the effectiveness of different policies in dynamic, unpredictable environments, industrial AI helps enterprises increase process efficiency, improve asset utilization, increase yields, and optimize the design and management of complex systems. Greater efficiency & productivity. Industrial AI lets enterprises enhance the results they achieve through automation, resulting in increased production, increased product quality, lower labor costs, reduced errors and rework, lower material consumption and less waste. These benefits correspond directly to the three broad opportunities for applying industrial AI that we introduced in its definition and expand on in the following section. Together, they enable enterprises to improve operational and business performance, while simultaneously increasing agility and innovation.

The industrial edge computing is also facing some new challenges. Firstly, distributed computing relies on connectivity. The industrial edge computing requires interoperability between various existing industrial fieldbuses as well as emerging technologies including time-sensitive network (TSN) [3], software-defined network (SDN) [4], 5G [5], etc. The existing industrial fieldbuses are not compatible with each other even defined in the IEC 61158 standard [26]. How to ensure the real-time constraints, scalability, and reliability of heterogeneous networks for industrial edge computing is a key problem that must be solved. Secondly, data mining is also enabled with industrial edge computing. How to perform effective data mining on edge nodes is also a big challenge. The ECNs as the data acquisition entry points usually generate massive real-time process data with milliseconds interval. These data could be used to restore entire operation that is valuable for innovative applications including product life cycle management, preventive maintenance, assets management, control optimization, etc. By adopting data optimization for edge computing, timeliness and validation of processed data can be assured for these industrial applications.

**CHAPTER – 2**

**LITERATURE REVIEW**

1. **LITERATURE REVIEW**

During 20th century a brief history of AI can be given as: 1923 – Karel Kapek’s play named “Rossum’s University Robots (RUR)” opens in London, first use of the word “robot” in English. 1945 – Isaac Asimov, alumni at Columbia University, invented the term Robotics. 1950 –Turing Test for evaluation of intelligence was introduced by Alan Turing. Claude Shannon published detailed Analysis of chess playing as a search. 1956 – John McCarthy coined the term Artificial Intelligence. 1958 – John McCarthy invents LISP programming language for AI. 1964 – Danny Bobrow’s thesis at MIT showed that computers can understand natural language well enough to solve algebra word problems correctly. 1979 – The First Computer controlled autonomous vehicle, Stanford Cart was built. 1984 – Dennett discusses the frame problem and how it relates to the difficulties arising from attempting to give robots common sense.

1990 – Major advances in all area of AI:

Significant demonstrations in Machine Learning

Case-based reasoning

Multi-agent planning

Scheduling

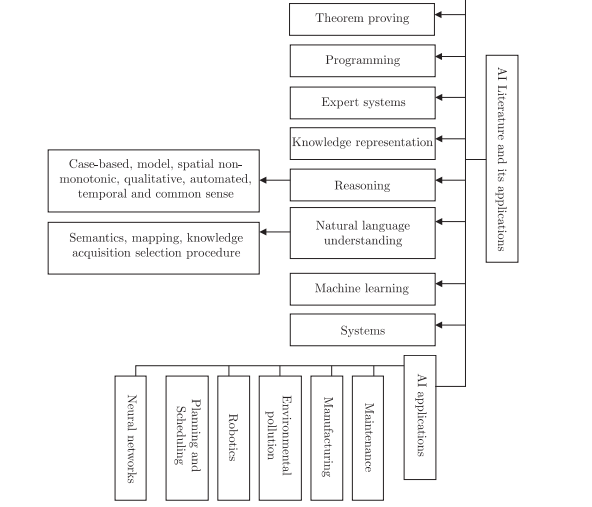
Data mining, web crawler

Natural Language understanding and translation

Vision, virtual reality

Games 1997 –

The Deep Blue Chess Program beats the World Chess Champion, Gerry Kasparov 2000 – Interactive Robot Pets become commercially available. MIT displays a robot with a face name – Kismet that expresses emotions.

  
 **Fig. 2.1** **Illustration concerning the relationship among the diverse fields of AI**

The two major approaches that has been developed for the regular AI system are: “top down” approach which started with the higher level functions and implemented those, and the “bottom up” approach which looked at the neuron level and worked up to create higher level functions.

In the 21st century artificial intelligence (AI) has become an important area of research in virtually all fields: engineering, science, education, medicine, business, accounting, finance, marketing, economics, stock market and law, among others (Halal (2003), Masnikosa (1998), Metaxiotis et al. (2003), Raynor (2000), Stefanuk and Zhozhikashvili (2002), Tay and Ho (1992) and Wongpinunwatana et al. (2000)).

The field of AI has grown enormously to the extent that tracking proliferation of studies becomes a difficult task (Ambite and Knoblock (2001), Balazinski et al. (2002), Cristani (1999) and Goyache (2003)). Apart from the application of AI to the fields mentioned above, studies have been segregated into many areas with each of these springing up as individual fields of knowledge (Eiter et al. (2003), Finkelstein et al. (2003), Grunwald and Halpern (2003), Guestrin et al. (2003), Lin (2003), Stone et al. (2003) and Wilkins et al. (2003)).

**2.1. The challenge of the AI field:**

This work grew out of the challenges that AI possesses in view of the rise and growing nature of information technology worldwide that has characterised business- and non-business organisational development (Barzilay et al. (2002), Baxter et al. (2001), Darwiche and Marquis (2002), Gao and Culberson (2002), Tennenholtz (2002) and Wiewwiora (2003)). The necessity for research in AI is being motivated by two factors that are

1. To give the new entrants into the AI field an understanding of the basic structure of the AI literature (Brooks (2001), Gamberger and Lavrac (2002), Kim (1995), Kim and Kim (1995), Patel-Schneider and Sebastiani (2003) and Zanuttini (2003)). As such, the literature discussed here answers the common query, “why must I study AI?”
2. The upsurge of interest in AI that has prompted an increased interest and huge investments in AI facilities. Interested researchers from all disciplines wish to be aware of the work of others in their field, and share the knowledge gleaned over the years (Rosati (1999), Kaminka et al. (2002), Bod (2002), Acid and De Campos (2003), Walsh and Wellman (2003), Kambhampati (2000) and Barber (2000)). By sharing AI knowledge, new techniques and approaches can be developed so that a greater understanding of the field can be gained. To these ends, this paper has also been written for researchers in AI so they can continue in their efforts aimed at developing this area of concentration through newly generated ideas. Consequently, they would be able to push forward the frontier of knowledge in AI. In the section that follows this paper presents a brief explanation of some important areas in Artificial Intelligence. This is to introduce the readers into the wide-ranging topics that AI encompasses. In another section, a comprehensive review of the literature along the major categories of artificial intelligence is presented. The review raises some important questions with serious research implications for those who are interested in carrying out research artificial intelligence. These questions if well addressed will solve some unresolved technical and non-technical issues carried over from the last decade to the present time.

**2.2. An overview of the AI field:**

On a very broad account the areas of artificial intelligence are classified into sixteen categories (Becker et al. (2000), Singer et al. (2000), Chen and Van Beek (2001), Hong

(2001) and Stone et al. (2001)). These are: reasoning, programming, artificial life, belief revision, data mining, distributed AI, expert systems, genetic algorithms, systems, knowledge representation, machine learning, natural language understanding, neural networks, theorem proving, constraint satisfaction, and theory of computation (Peng and Zhang (2007), Zhou et al. (2007) and Wang et al. (2007)).

Since many readers of this article may require a glance view of the AI field, the author has utilised a flow diagram to illustrate the whole structure of this paper, and the relationship among the diverse fields of AI, as presented in Figure 1. What follows is a brief discussion of some of the important areas of AI (Chan and Darwiche (2002), Pool and Zhang (2003), Bhattacharyya and Keerthi (2001), Chawla et al. (2002), Al-Ani and Deriche (2002) and Xu and Li (2000)). These descriptions only account for a selected number of areas.

**2.2.1 Reasoning:**

The first major area considered here is that of reasoning. Research on reasoning has evolved from the following dimensions: case-based, non-monotonic, model, qualitative, automated, spatial, temporal and common sense.

For an illustrative example, the case-based reasoning (CBR) is briefly discussed. In CBR, a set of cases stored in a case base is the primary source of knowledge. Cases represent specific experience in a problem-solving domain, rather than general rules. The main activities when solving problems with cases are described in the case-based reasoning cycle. This cycle proposes the four steps: relieve, reuse, revise and retain.

First, the new problem to be solved must be formally described as a case (new case). Then, a case that is similar to the current problem is retrieved from the case base. The solution contained in this retrieved case is reused to solve the new problem with a new solution obtained and presented to the user who can verify and possibly revise the solution. The revised case (or the experience gained during the case-based problem solving process) is then retained for future problem solving. Detailed information on “dimensions” or how they are related could be obtained from the relevant sources listed in the references (Debruyne and Bessiere (2001), Halpern (2000), Halpern (2001), Renz and Nebel (2001), Singh et al. (2002) and Straccia (2001)).

**Expert system:**

The third aspect of AI discussed here is expert system. An expert system is computer software that can solve a narrowly defined set of problems using information and reasoning techniques normally associated with a human expert. It could also be viewed as a computer system that performs at or near the level of a human expert in a particular field of endeavour.

**Natural language understanding:**

Natural language generation (NLG) systems are computer software systems that produce texts in English and other human languages, often from non-linguistic input data. NLG systems, like most AI systems, need substantial amounts of knowledge that is difficult to acquire. In general terms, these problems were due to the complexity, novelty, and poorly understood nature of the tasks the systems attempted, and were worsened by the fact that people write so differently (Reiter et al. (2003)).

**Knowledge representation (KR):**

Knowledge bases are used to model application domains and to facilitate access to stored information. Research on KR originally concentrated around formalisms that are typically tuned to deal with relatively small knowledge base, but provide powerful reasoning services, and are highly expressive.

**Reasoning in artificial intelligence:**

The theory and practice of reasoning in artificial intelligence has extensive documentation (Atkinson and Bench-Capon (2007)). Researchers have worked in terms of:

1. development of axioms that give sound and complete axiomazation for the logic of reasoning;
2. the theoretical properties of the algorithms used for qualitative temporal
3. what is relevant to a given problem of reasoning (independence);
4. A study on axomatising causal reasoning is credited to Halpern (2000).
5. The author axomatised causal models defined in terms of a collection of
6. equations as defined by Pearl. Axiomatisations are provided for three
7. successively more general classes of causal models
8. the class of recursive theories (those without feedback);
9. the class of theories where the solutions to the equations are unique;
10. arbitrary theories (where the equations may not have solutions and, if they do, they are not necessarily unique). It is shown that to reason about causality in the most general third class, we must extend the language used by Galles and Pearl.

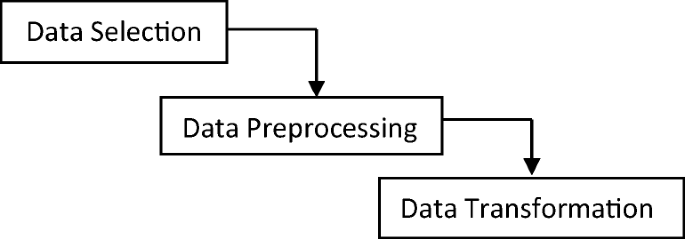
In addition, the complexity of the decision procedures is characterised for all the languages and classes of models considered. The concept of reasoning in Artificial Intelligence has been discussed under some general areas, which include complexity of reasoning, reasoning about minimal belief, axiomatising, sampling algorithm, conditional plausibility, efficient methods, logic and consistency, fuzzy description logics, backbone fragility, diagnosis, independence, domain filtering, and fusion. The literature on complexity of reasoning relates to spatial congruence and expressive description logics. Cristani (1999) introduces a novel algebra for reasoning about spatial congruence, thus, showing that the satisfiability problem in the spatial algebra MC-4 is NP-complete, and present a complete classification of tractability in the algebra, based on the individuation of three maximal tractable sub classes, one containing the basic relations.

**CHAPTER – 3**

**METHODOLOGY**

1. **METHODOLOGIES**

Existing systems uses sensors which are monitored and controlled only based on the data received from the industrial site. But, the data will be cleared in servers after the completion of action. As previous data or values are not stored in the system again if any uneven situation occurs again it need sense the conditions and store the data. In this paper we are proposing a framework for data preparation. In first step, the data selection considers what data is presented, what data is misplaced and what data can be deleted. In second step the Preprocessing of data organizes formatting the selected data, cleaning and sampling from it. Final step supports transformation of data; it transforms preprocessed data ready for ML by engineering features using scaling, attribute decomposition and attribute aggregation as shown in fig. [3](https://link.springer.com/article/10.1007/s42797-020-00020-y#Fig3).1.

[](https://link.springer.com/article/10.1007/s42797-020-00020-y/figures/3)

**Fig. 3.1 Three step framework for data preparation**

Data preparation is a huge subject that can engage many iterations, searching and analysis. Good data preparation is a challenging task in machine learning. There is an important requirement that one should notify the actual data require to address the problem working on it. Future assumptions should record carefully, so that test can be done later if required. Data selection step select the subset of all existing data that will be working on it. There is always a strong desire for including all data “more is better” is available will store. Next step is Preprocess Data, it describes that after data selection, and the main concern is how to use the data efficiently. This step is mainly for formatting the selected data to solve the identified problem. Data preprocessing steps commonly used are formatting, cleaning and sampling. Formatting the data is put it into a relational database model and it would be in a specified format. Cleaning data means removal or fixing of missing data. Sampling means, usage of one part of main dataset for clarifications or experimentations, ones it successes, working on the whole data set.

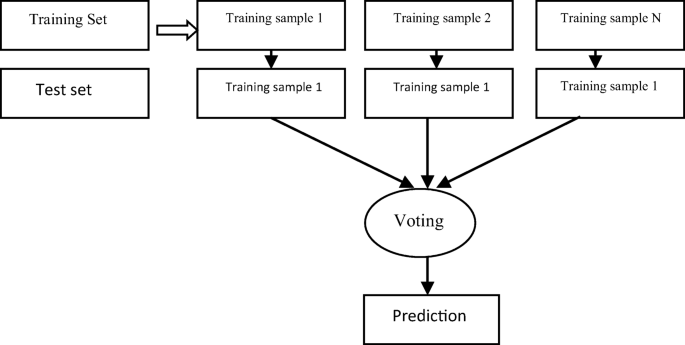
Typically machine learning tools used data will influence the preprocessing. The last step is data transformation, which influence the specific algorithm and likely have to revisit different transformations of preprocessed data and work on the problem. This is called feature engineering. And it can be very advantageous to increase the performance of an algorithm. ML techniques are used to develop a system which will automatically monitor the industrial safety applications, sensors, and detecting the uneven conditions and generate Alerts/Alarms basing on previous history to take intelligent decision or crucial decision by controlling devices within industries.

One of the supervised learning algorithms is Random forest algorithm, which is also used for classification, and it creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It reduces the over-fitting by averaging the result. Random forest algorithm starts with the selection of random samples from a given dataset shown in Table [1](https://link.springer.com/article/10.1007/s42797-020-00020-y#Tab1). And construct a decision tree for every sample. Then it will get the prediction result from every decision tree. In the next step, voting will be performed for every predicted result. Finally select the most voted prediction result as the final prediction result.

### Application of random forest algorithm

First step starts with the selection of random samples from a dataset (Table [1](https://link.springer.com/article/10.1007/s42797-020-00020-y#Tab1)), after that algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree. Finally voting will be performed for every predicted result and most voted prediction result set as the final prediction result. Figure [4](https://link.springer.com/article/10.1007/s42797-020-00020-y#Fig4) illustrates the working of Random Forest Algorithm. Data Preprocessing will be done with the help of following script lines.

* X = dataset.iloc[:,: -1] - TRAINING DATA SET values,
* y = dataset.iloc[:, 4] - TESTING DATA SET values
* Next, we will divide the data into train and test split. The following code will split the dataset into 70% training data and 30% of testing data −
* from sklearn.model\_selection import train\_test\_split
* X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size) – IMPORT FUNCTION FOR SCRIPTING LINES

[](https://link.springer.com/article/10.1007/s42797-020-00020-y/figures/4)

**Fig. 3.2 Working of Random Forest Algorithm**

Next, train the model with the help of Random Forest Classifier class of sklearn as follows:

Formulae: sklearn.ensemble import Random Forest Classifier.

### Main function of algorithm

* classifier = RandomForestClassifier(n\_estimators)
* classifier.fit(X\_train, y\_train)
* At last, we need to make prediction. It can be done with the help of following script −
* y\_pred = classifier.predict(X\_test)

### Working process of an algorithm

1. Step 1:

First, start with the selection of random samples from a given dataset.

It means Pick N random records from the dataset.

X = dataset.iloc[:,: -1] -TRAINING.

y = dataset.iloc[:, 4]-TESTING

1. Step 2:

Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.

Build a decision tree based on these N records.

Choose the number of trees in algorithm and repeat steps 1 and 2.

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size)-SPLITING THE TREE

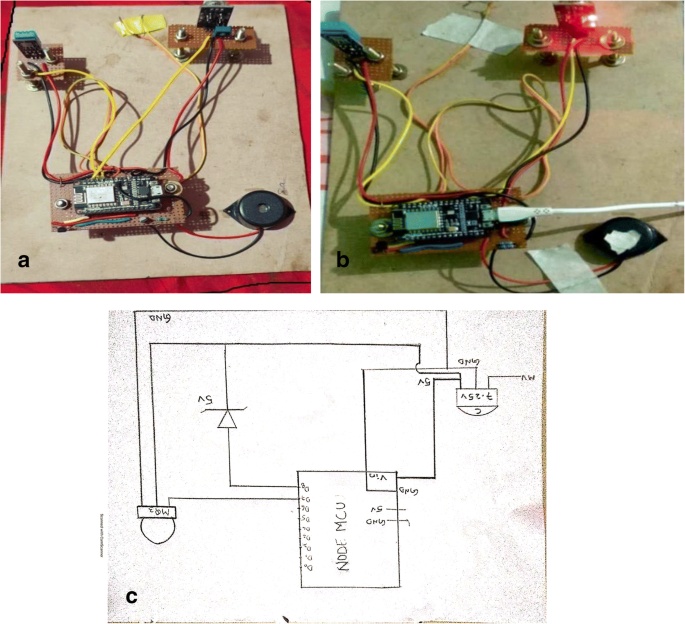
1. Step 3:

In this step, voting will be performed for every predicted result.

In case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

y\_pred = classifier.predict(X\_test).

The industrial automation safety system was designed to improve the efficiency and accuracy of sensors. The goals and objectives were based on the issues that focused on sensors in industry. Proposed system uses software and hardware components with IOT and machine learning algorithm. We collect the sensors data set of one year and train the respective model shown in figs. [5(a)](https://link.springer.com/article/10.1007/s42797-020-00020-y#Fig5) and [5(b)](https://link.springer.com/article/10.1007/s42797-020-00020-y#Fig5). To develop a system which will automatically monitor the industrial applications, sensors, and detecting the uneven conditions and generate alerts basing on previous history to take intelligent decision by controlling devices within industries using the concept of machine learning. Where we store the data and monitor the sensor and upload information to users. Based on that data it will give notifications or generate alerts again if any hazards take place in industry.

**Fig. 3.3**[](https://link.springer.com/article/10.1007/s42797-020-00020-y/figures/5)

(a) Laboratory prototype of IoT based system model

(b) IoT based system model with alarm

(c) Block diagram of Node mcu circuit

Things are automatically assigned to deploy, manage, and schedule the behavior and timely switching as per requirements. These enable things and devices to perform task in real time automatically. It measure the distance of target objects or materials through the smart ML process and measure the distance without damage from machines to worker. Using this system one can operate the IoT from anywhere easily. The system will not only raise alarms but it also sends notification to mobile with danger warning whenever it detect odd situation or odd values in the industrial parameters.

### System modules

Proposed model focuses on the safety and accurate control of IOT sensors. The system model of our scheme consists of user authentication, IOT parameters (sensors), training dataset, and testing dataset.

### User authentication

First user need to register into the blynk application with respective hotspot id and password. It is a Platform with IOS and Android apps to control Arduino, Raspberry Pi. It’s a digital dashboard where you can build a graphic interface by simply dragging and dropping widgets. It will connect the hardware to respective software.

### IOT parameter system

It is device of different sensors namely smoke, gas, temperature, humidity and fire with node mcu chip and buzzer shown in figs. [5](https://link.springer.com/article/10.1007/s42797-020-00020-y#Fig5) (a) and 5 (b). It is designed in order to detect the odd situation or danger situation in industry and address the user before so that they can control the accidents in industry. It has node mcu chip which is previously embedded with c code to connect with software shown in below fig.

### Training dataset

The model is initially fit on a training dataset, which is a set of examples used to fit the parameters of the model. The model is trained on the training dataset using a supervised learning method. In practice, the training dataset often consists of pairs of an input vector and the corresponding output vector, which is commonly denoted as the target. The current model is run with the training dataset and produces a result, which is then compared with the target, for each input vector in the training dataset. Based on the result of the comparison and the specific learning algorithm being used, the parameters of the model are adjusted. The model fitting can include both variable selection and parameter estimation. Train the classifier using training set, it has values which are stored previously so instead of writing code what we do is using these values we train the model repeatedly with help of random forest algorithm. It is the main module, and without the training dataset we cannot get the result.

### Testing datasets

Finally, the test dataset is a dataset used to provide an unbiased evaluation of a final model fit on the training dataset. If the data in the test dataset has never been used in training (for example in cross-validation), the test dataset is also called a holdout dataset.

**3.1 Industrial automation using RPA**

The automation of robotic processes has been experiencing an increasing trend of interest in recent times. However, most of literature describes only theoretical foundations on RPA or industrial results after implementing RPA in specific scenarios, especially in finance and outsourcing. This paper presents a systematic mapping study with the aim of analyzing the current state-of-the-art of RPA and identifying existing gaps in both, scientific and industrial literature. Firstly, this study presents an in-depth analysis of the 54 primary studies which formally describe the current state of the art of RPA. These primary studies were selected as a result of the conducting phase of the systematic review. Secondly, considering the RPA study performed by Forrester, this paper reviews 14 of the main commercial tools of RPA, based on a classification framework defined by 48 functionalities and evaluating the coverage of each of them. The result of the study concludes that there are certain phases of the RPA lifecycle that are already solved in the market. However, the Analysis phase is not covered in most tools. The lack of automation in such a phase is mainly reflected by the absence of technological solutions to look for the best candidate processes of an organization to be automated. Finally, some future directions and challenges are presented.

Robotic Process Automation is the technology that allows anyone today to configure computer software, or a “robot” to emulate and integrate the actions of a human interacting within digital systems to execute a business process. RPA robots utilize the user interface to capture data and manipulate applications just like humans do. They interpret, trigger responses and communicate with other systems in order to perform on a vast variety of repetitive tasks. Only substantially better: an RPA software robot never sleeps and makes zero mistakes.

An HR service provider from Europe was processing 2,500 sick leave certificates per month with an average handling time of four minutes per item. Within three weeks they implemented an RPA solution and achieved 90% process automation. The RPA robot extracts data from a transaction in SAP, inserts the information into the customer’s systems, and prints it. The HR service provider achieved a return-on-investment within six months, with error rates reduced to 0%, manual effort reduced to 5%, and processing time reduced by 80%.

A global retailer was using its store closing reports to validate closing information for each of its registers across hundreds of stores. The store’s employees used a manual and sluggish process to pull up these reports. By automating the process the store freed up its employees to now focus on more customer-centric activities. The RPA robots now move the closing reports to one server, then read and consolidate the needed information for the store’s closing reports.

A trade credit insurance company with over 50,000 clients worldwide automated the credit limit request underwriting process. Underwriters were previously gathering information manually, from internal (Risk & Policy) to external (Customer Site, Google News) sources. With RPA, they saved 2,440 hours of human work a month. Employees now use that time to work directly with customers.

In this context, it is possible to identify and group different ideas and proposals to allow continuous improvement using a complete RPA lifecycle. This lifecycle is used to compare and categorize the primary studies identified in this SMS. Then, the RPA lifecycle has the following phases:

Analysis Phase. This phase consists of analyzing and determining the viability of carrying out the automation of a certain process by means of a detailed analysis of the effort involved in the self-motivation of such process considering the execution characteristics of the process itself.

* Design Phase. The process design phase begins for those processes that have passed the previous feasibility analysis. The purpose of this phase is to detail the set of actions, data flow, activities, etc., that must be implemented in the RPA process. • Construction Phase. This phase consists of implementing each of the automatable parts of each process identified in the design phase.
* Deployment Phase. The robots obtained as a result of the construction phase need an environment in which to be executed, just as a human operator needs an environment in which to perform his work. This environment, in the context of RPA, usually corresponds to a computer that has an installation of one or more information systems. Each robot must be executed in its own execution environment since the replacement between human operator and software is direct.
* Control and Monitoring Phase. Once the robots are deployed in their respective execution environments, this phase oversees controlling and monitoring the performance of each robot. In this phase, the execution of robots is launched, it stops in case of serious errors, the execution status is monitored, etc., until they have finished their work. • Evaluation and Performance Phase. The last phase of the process consists of the evaluation of the robots’ performance.

Finally, it is interesting to mention that, although there are promising researches in RPA published in the last decade (for instance researches that are compiled in this systematic review) in different business environments, today it is possible to glimpse a promising future. Devarajan [24] argues a great growth in the application of RPA in companies because of the growth in unstructured data, repetitive business tasks and evolution of new business processes, among other factors. RPA’s future is geared towards the significant improvement of the quality, operational scalability and productivity of employees through integration with cognitive technologies (such as Artificial Intelligence or Machine Learning) and its integration with structured, unstructured and semi-structured data, natural language processing capability to enhance human interaction and skills to adapt.

The aim of this work is to presents a three-real-time architecture based on edge computing. We hope that by putting forward this concept can arouse widespread concern in the research of three-real-time architecture. By sorting out the literature of edge computing, industrial automation and artificial intelligence, this paper puts forward the problems existing in the process of intelligent transformation and upgrading of industrial automation industry. In the future, we will build a Three-Real-Time platform, a network engineering research center for edge computing. To improve the real-time algorithm, the virtualization of industrial automation controller, and the optimization of embedded artificial intelligence inference.

Industrial automation implemented on deep learning includes effective architectural approach

The Two level hybrid architecture with Robotic Process Automation used with Artificial intelligence to improve the real-time algorithm, so that the virtualization of industrial automation controller, and the optimization of embedded artificial intelligence in a better and efficient way.

## Applications of AI in various industries

The influence of AI technology can be seen across sectors such as transportation, education, manufacturing, online shopping, communication, sports, media, healthcare, politics and government, banking and finance, aerospace, and so much more.

Below is a list of essential industries impacted by AI:

* **Transportation:** Autonomous car, also known as a self-driving car, is a vehicle that can sense its environment and is capable of moving without human interference. This technology can transform the transportation system, because it can analyze traffic and alternative routes, thus reducing travel times.
* **Manufacturing:** high performing robots work faster, and complete tasks more efficiently than humans. Also, they can work for long periods nonstop as long as the power required for them to function is available. By using 3D technology and machine vision, these machines can speed up the process of product manufacturing.
* **Healthcare:** Applications such as autonomous surgical robots, virtual nursing assistants, automated image diagnosis, and dosage error reduction have been some of the ways AI has been crucial for the technological advancements in the health sector.
* **Entertainment:** machine learning can predict a user’s behavior to make recommendations on the type of movie, music, TV shows, and other content he’ll be interested in. Also, adverts can now be personalized based on the user’s preference, thereby increasing the chances a marketer will make a sale.
* **Sports:** AI technology like automation and predictive analysis can be used in business decisions, sponsorship activations, ticket sales, and determining athletes’ performance.

Future applications of AI would be utilized in automated transportation, cyborg technology, solving problems associated with climate change, deep-sea and space exploration.

If the [projected growth of the AI software market](https://techjury.net/blog/ai-stats/) from $1.4 billion in 2016 to $59.8 billion in 2025 is anything to go by, AI is set for a massive takeover in the coming years.

Industrial automation implemented on deep learning includes effective architectural approach. The Two level hybrid architecture with Robotic Process Automation used with Artificial intelligence to improve the real-time algorithm, so that the virtualization of industrial automation controller, and the optimization of embedded artificial intelligence in a better and efficient way.

**1ST LAYER: REAL-TIME HARDWARE NETWORK**

Real-Time Ethernet (RTE) classification structure could be based on the delivery time [13]. There are three classes:

Human control class: The delivery time is around100 ms. It belongs to low-speed control. This time requirement is a typical visual pause for human beings. This type can be used for data presentation in SCADA system. It can be fulfilled with TCP/IP communication channel.

Process control class: The delivery time is below 10ms. This is the requirement for most production line ,machine tool control system etc. Field bus has been the mainstream for a long time until Industrial Ethernet has emergence. This type can be used for programmable logic controllers (PLCs) or PC-based control.

Motion control class: The delivery time is below 1ms. It can be to synchronize several axes over a network, both protocol and hardware structure are modified, such as EtherCAT [14]. Motion controller is specially used in stepper motor and servo motor. High speed and high precision are its characteristics. So motion control is the most advanced test of real time network. Because of the speed and certainty of signal transmission, RTE can be used on the terminal side only with I/O, all communication functions of hardware without processors or software to execute. That is to say, it can focus on edge computing side programming and without losing the high speed real-time characteristics.

RTE can replace the traditional mainstream Field bus with wired or wireless mode. So in this way, not only can it was easily connected with the second layer of real-time operating systems, but also easily connected with traditional industrial automation equipment. At present, the way to upgrade industrial automation intelligently is to add industrial real time Internet network to PLC control module to achieve high-speed linkage control. However, because of the existence of the CPU module controlled by PLC, the standard is not uniform and the way is diversified, different components of multiple suppliers can not be used in the same architecture. In fact, depending on the deterministic high-speed RTE network, it is possible to remove the CPU module of the field PLC and migrate the control logic programs to the upper layer. Obviously, the real-time network performance of hardware platform is the key to ensure the successful migration of the control logic programs to the upper layer. If migration is successful, a new application of software defined industrial control will emerge. This is the key of unified programming method of IEC61131-3 standard. On the physical hardware side, as shown in figure III, only I/O devices are left, no CPU Module, and of course.

**2ND LAYER: REAL-TIME OPERATING SYSTEM**

In general, an operating system (OS) is responsible for managing the hardware resources of a computer and hosting applications that run on the computer, Real-time is not so important in non-industrial situations. Real-Time Operating System (RTOS) [24] is specially designed to meet rigorous time constraints. Most RTOS is often present in embedded systems, and users will not notice them in most cases. RTOS is an operating system with temporal and spatial partitioning.

RTOS must have a known maximum time for each of the critical operations. RTOS has two kinds: hard real-time and soft real-time. This hard real-time can absolutely guarantee a maximum time for these operations. Soft real-time may lose a small amount of data occasionally. So only soft real-time is needed in data polling acquisition, but when issuing instructions in time synchronization, hard real-time should be adopted.

Operating systems like Windows are designed to maintain user responsiveness with many programs, and are not the ideal platform for running applications that require precise timing or extended up-time. But real-time operating systems are designed to run critical applications reliably with precise and predictable timing. If programmed correctly, an RTOS can guarantee that a program will run with very consistent timing, and checking to make sure that important deadlines are met. In this paper, RTOS serves as a link between the preceding and the following.

Since 2016, I put forward ICICOS, my team planned to build software PLC technology based on the IEC61131-3 standard of the virtual cloud computing side. We hope to rely on software to realize the function of PLC and add callable AI program block to realize intelligent control. In the first generation of ICICOS system, we built a cloud computing platform based on Docker container technology, it's built on a host with the Ubuntu operating system [25]. We refer to Sousa's code[26] and make some modifications, a visual program method of Docker Container is Successfully Implemented and Connecting with physical input and output devices through TCP/IP Modbus Protocol [27]. But there are uncertainties and delays in this system.

In this layer, development a professional RTOS based on ICICOS is an important task. We need to migrate it to a new environment, and rule the law of RTOS schedule. This is the hardware and software system, which has been embedded in the application of product and engineering, and is cored with MCU or ARM. Fortunately, there are some combs research.

**CHAPTER – 4**

**RESULTS &COMPARISON**

1. **RESULTS AND COMPARISON**

Here we consider the sensor data set of 1 year which consists of 40 to 50 data entries. By using the threshold values of each parameters like humidity, temperature, fire, gas, smoke etc., we decide whether to rise alarm or not. Print the values of room temperature and humidity value, fire degree, gas and smoke value and send notifications to the respected mobile to which the hardware kit is connected. Now first task is to connect hardware kit to the mobile application called blank using our portable hotspot user id and password. After connecting the kit to our application show notification as IOT parameters are connected and device is online. Now code through which we have trained data sets is executed and the data sets will be in process. Run the code in the python IDLE after running that we run the c code in Arduino IDE. Now we will keep the smoke, fire in front of the sensors it will detect those parameters and send the values to the ML. Now ML based on the previous values it will analyze weather it is having high values or low values. The code is implemented using the proposed algorithm. It is a decision tress based algorithm which will take multiple values and give single value in terms of 0’s and 1’s.

Random forests are very flexible and possess very high accuracy. Scaling of data does not require in random forest algorithm. It maintains exactness even after providing data without scaling. This algorithm maintains accuracy even a huge quantity of the data is missing.

**4.1 RPA different from other enterprise automation tools**

How is RPA different from other enterprise automation tools?  
 In contrast to other, traditional IT solutions, RPA allows organizations to automate at a fraction of the cost and time previously encountered. RPA is also non-intrusive in nature and leverages the existing infrastructure without causing disruption to underlying systems, which would be difficult and costly to replace. With RPA, cost efficiency and compliance are no longer an operating cost but a byproduct of the automation.

**4.2 RPA works**

How does Robotic Process Automation work?  
 RPA robots are capable of mimicking many–if not all–human user actions. They log into applications, move files and folders, copy and paste data, fill in forms, extract structured and semi-structured data from documents, scrape browsers, and more.

An HR service provider from Europe was processing 2,500 sick leave certificates per month with an average handling time of four minutes per item. Within three weeks they implemented an RPA solution and achieved 90% process automation. The RPA robot extracts data from a transaction in SAP, inserts the information into the customer’s systems, and prints it. The HR service provider achieved a return-on-investment within six months, with error rates reduced to 0%, manual effort reduced to 5%, and processing time reduced by 80%.



**Fig. 4.1** **Content distribution network (CDN)**

Content distribution network (CDN) [31] is an effective approach to improve internet service quality, CDN replicates the content from the place of origin to the replica servers.The development of virtual machine (VM) and CDN forms Cloud Computing. Edge Computing has gradually emerged based on VM-Based Mobile Computing [32]. In fact, Edge Computing is virtualization of embedded systems. If we use multiple RTOS to construct virtual computing, it will be able to meet the overall real-time requirements of the system, which is the Real-time Edge Computing. As shown in Figure Using CDN and embedded virtualization technology, the scattered real-time control system can be integrated under the same platform. Top applications can be scheduled under this platform.

**Real-time Edge Computing:**

In order to overcome the scalability problem of the traditional Industrial Automation architecture, Figure can be considered as a dynamic and scalable platform. It can be handling the data streams at the edge. All hardware is connected by RTE, it only enhances the computing power of the computing resource pool. Like cloud computing, people don't even know where its physical input and output interfaces are. If a GPU server is added to the edge, the ability of deep learning and training will be enhanced.

Real-time Edge Computing should have the following characteristics:

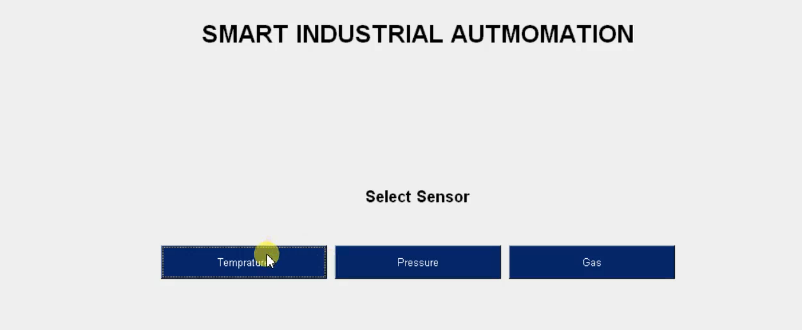
Certainty: Including the certainty of communication signals, the certainty of computing tasks.

Extensibility: If too many computational tasks leads to real-time performance degradation, computing power can be increased dynamically.

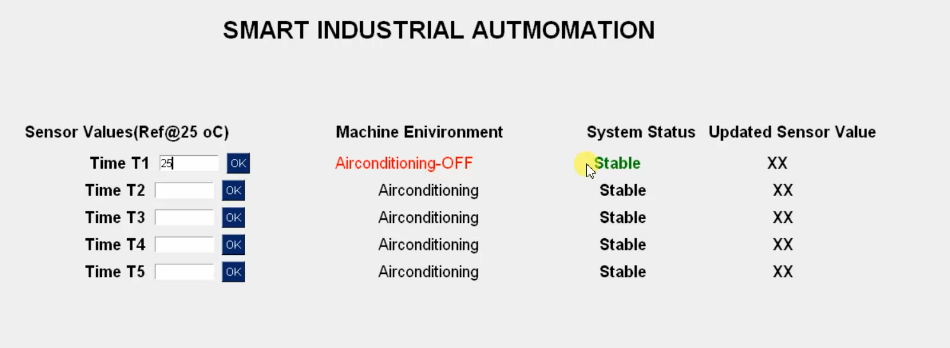
**CHAPTER -5**

**OUTPUTS**

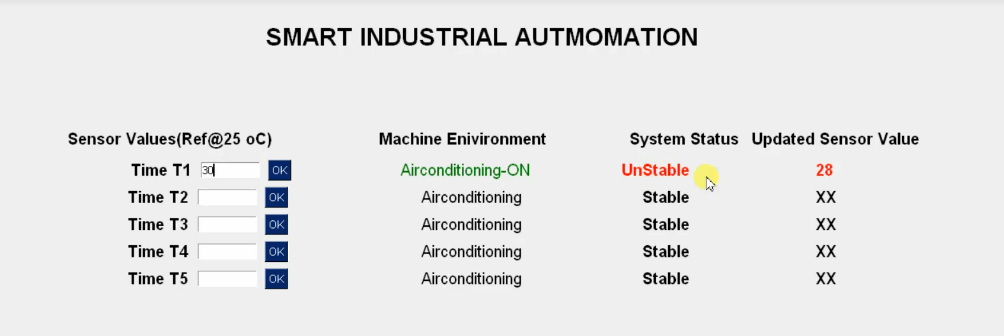
1. **OUTPUTS**

****

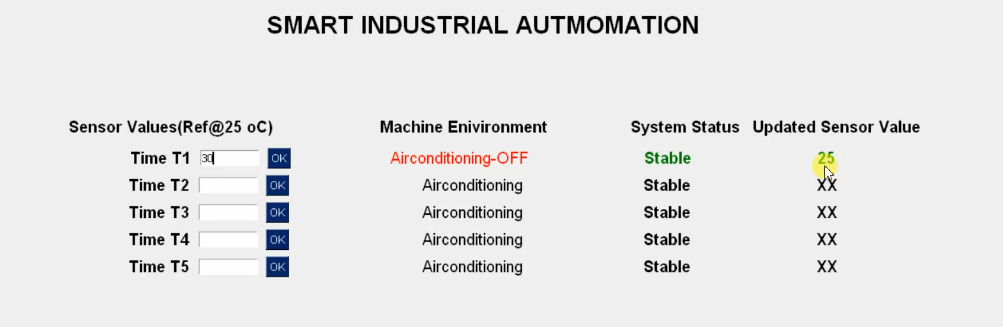
**Fig.no.5.1** Screenshot on selecting sensor temperature

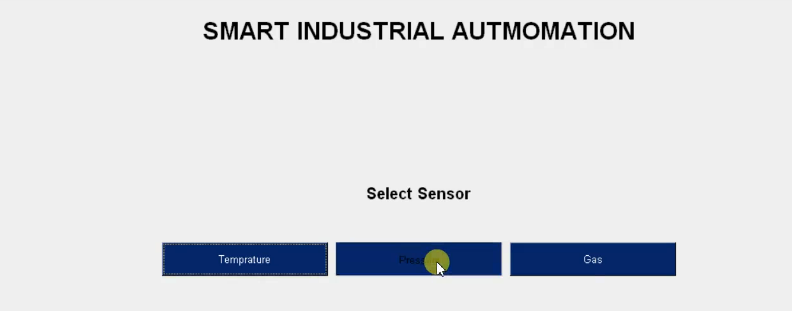
****

**Fig.no.5.2** Screenshot on selecting temperature 25oC

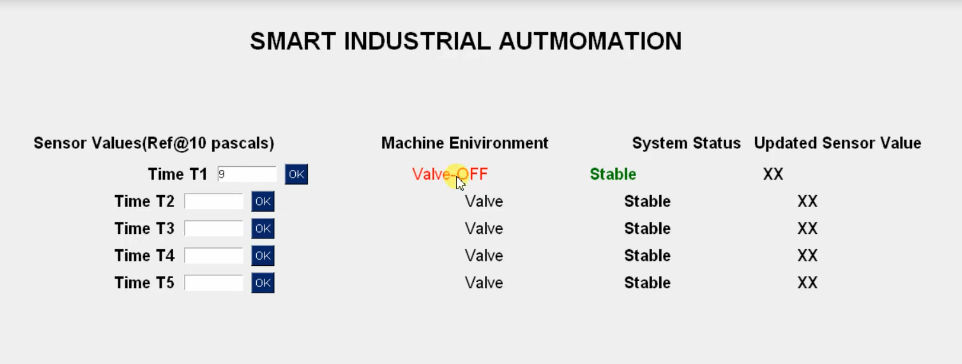
****

**Fig.no.5.3**  Screenshot on selecting temperature 30oC

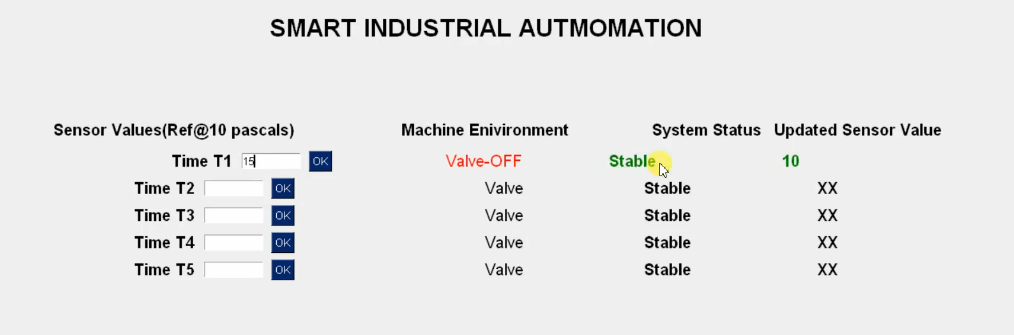
****

****

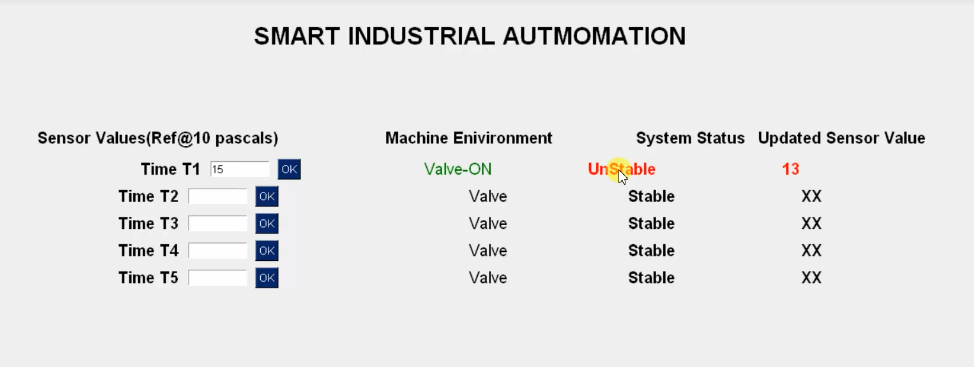
**Fig.no.5.4** Screenshot on selecting sensor pressure

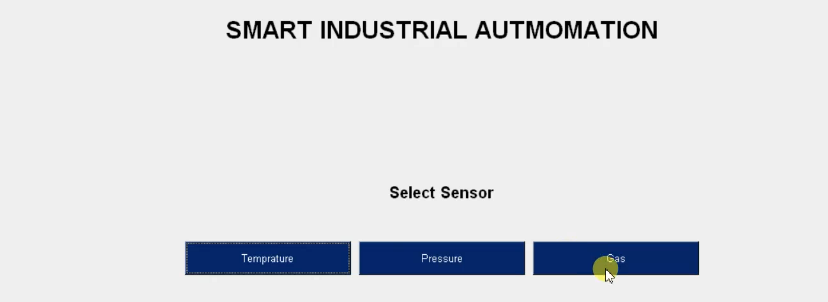
****

**Fig.no.5.5** Screenshot on selecting pressure 9Pascals

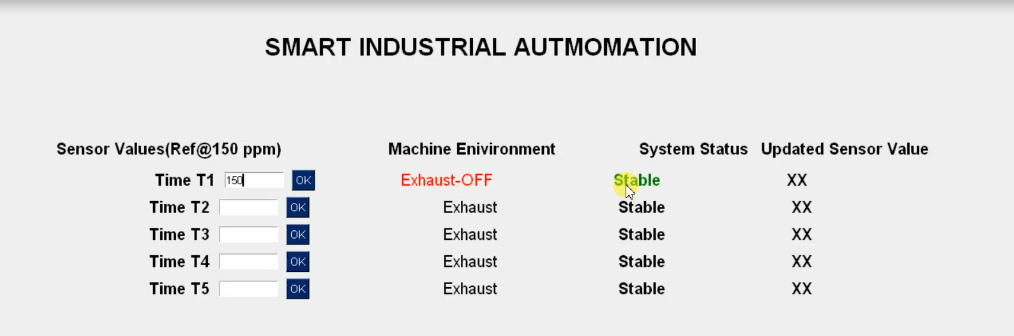
****

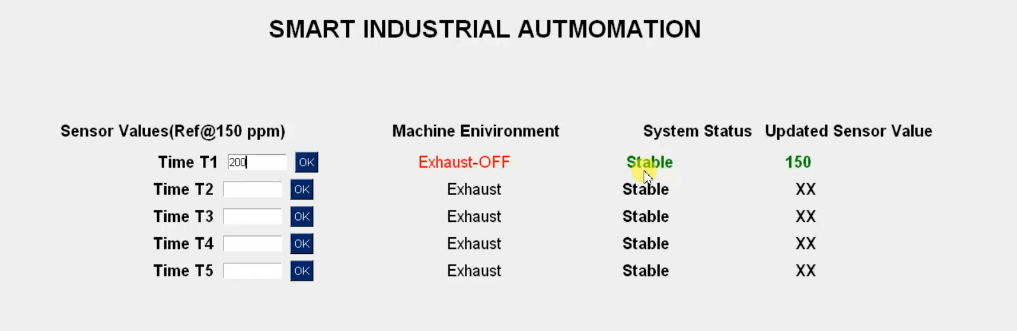
**Fig.no. 5.6**  Screenshot on selecting pressure 15pascals

****

****

**Fig.no.5.7** Screenshot on selecting sensor gas

**Fig .no.5.8** Screenshot on selecting gas 150ppm

****

**Fig.no.5.9** Screenshot on selecting gas 200ppm

**CHAPTER -6**

**CONCLUSION**

1. **CONCLUSION**

Existing systems uses sensors which are monitored and controlled only based on the data received from the industrial site. But, the data will be cleared in servers after the completion of action. As previous data or values are not stored in the system again if any uneven situation occurs again it need sense the conditions and store the data. System sometimes cannot give the accurate results because sensor cannot differentiate things like human. It may fail to take important decisions and give false alerts. The proposed system is to prevent accidents and control appliances in the industry. This system is capable for detecting fire, heat, various hazard gases and uneven conditions by providing the location of the affected region and generates alarms and control machines using the concept of IOT.

The proposed system is trained repeatedly using the concept of machine learning. Where we store the data and monitor the sensor and upload information to users. Based on that data or previous history it will give notifications or generate alerts if any hazards take place in industry. System is able to differentiate things like human and also capable of taking crucial decisions. It will give accurate results. In this paper we proposed a framework for data preparation to machine learning algorithm. Adopted random forest algorithm possesses flexibility and high accuracy. Here we proved that accuracy is 0.97 from Table [2](https://link.springer.com/article/10.1007/s42797-020-00020-y#Tab2), so we can provide safety aspects in industrial automation with the proposed methodology. This paper extending the use of random forest algorithm to fulfill the purpose of reduce false deduction, take important decisions and to get accurate results in very short period of time. It will raise alarms besides it will send notification to the mobile with danger message before accidents occur.

The aim of this work is to presents a two level hybrid architecture based on edge computing. We hope that by putting forward this concept can arouse widespread concern in the research of three-real-time architecture. By sorting out the literature of edge computing, industrial automation and artificial intelligence, this paper puts forward the problems existing in the process of intelligent transformation and upgrading of industrial automation industry. In the future, we will build a Three-Real-Time platform, a network engineering research center for edge computing. To improve the real-time algorithm, the virtualization of industrial automation controller, and the optimization of embedded artificial intelligence inference.

**CHAPTER -7**

**REFERENCES**

**7. References**

[1] Acid, S. and De Campos. L. M., Searching for bayesian network structures in the space of restricted acyclic partially directed graphs, Journal of Artiﬁcial Intelligence Research, Vol. 18, pp.445-490,2003.

[2] Alai, M., AI, scientiﬁc discovery and realism, Minds and Machines, Vol. 14, No. 1, pp.21-42, 2004.

[3] Al-Ani, A. and Deriche, M., A new technique for combining multiple classiﬁers using the dempster-shafer theory of evidence, Journal of Artiﬁcial Intelligence Research, Vol. 17, pp.333-361, 2002.

[4] Ambite, J. L. and Knoblock, C. A., Planning by rewriting, Journal of Artiﬁcial Intelligence Research,Vol. 15, pp.207-261, 2001.

[5] Argamon-Engelson, S. and Dagan, I., Committee-based sample selection for probabilistic classiﬁers,Journal of Artiﬁcial Intelligence Research, Vol. 11, pp.335-360, 1999.

[6]. Allen, J.F.: Maintaining about Temporal Intervals. Comm. of the ACM, 1983 Val 26 Nr 11

[7]. Dorn, J.: Wissensbasierte Echtzeitplanung. Vieweg , 1989 Laffey, Th.J. et al: Real-Time Knowledge-Based Systems.

[8]. AI Maaazine, 1988 Spring, pp 27 - 45. Lauber, R.J: Forecasting Real-Time Behaviour During Software Design using a ASE Environ-Int. Conf. on System Science 1989, pp 645-653.

[9]. Lesser, V.R. and Pavlin, J. and Durfee, E.: Approximate Processing in Real-Time Solving. Maaazine, 1988 Spring, pp 49 - 61.

[10].Varghese, A.; Tandur, D. Wireless requirements and challenges in Industry 4.0. In Proceedings of the International Conference on Contemporary Computing and Informatics (IC3I), Mysore, India, 27–29 November 2014; pp. 634–638.

[11].Trakadas, P.; Nomikos, N.; Michailidis, E.T.; Zahariadis, T.V.; Facca, F.M.; Breitgand, D.; Rizou, S.; Masip-Bruin, X.; Gkonis, P. Hybrid clouds for data-intensive, 5G-enabled IoT applications: An overview, key issues and relevant architecture. Sensors 2019, 19, 3591.

[12].García-Magariño, I.; Muttukrishnan, R.; Lloret, J. Human-Centric AI for Trustworthy IoT Systems with Explainable Multilayer Perceptrons. IEEE Access 2019, 7, 125562–125574.

[13].Moura, R.; Ceotto, L.; Gonzalez, A.; Toledo, R. Industrial Internet of Things (IIoT) Platforms-An Evaluation Model. In Proceedings of the International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 12–14 December 2018; pp. 1002–1009.

[14].Zafeiropoulos, A.; Fotopoulou, E.; Peuster, M.; Schneider, S.; Gouvas, P.; Behnke, D.; Muller, M.; Bok, P.-B.; Trakadas, P.; Karkazis, P.; et al. Benchmarking and Profiling 5G Verticals’ Applications: An Industrial IoT Use Case. In Proceedings of the 6th IEEE Conference on Network Softwarization (NetSoft), Ghent, Belgium, 29 June–3 July 2020; pp. 310–318.

[15].Zahariadis, T.; Voulkidis, A.; Karkazis, P.; Trakadas, P. Preventive maintenance of critical infrastructures using 5G networks & drones. In Proceedings of the 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 29 August–1 September 2017.

[16].Fotiadou, K.; Velivassaki, T.; Voulkidis, A.; Railis, K.; Trakadas, P.; Zahariadis, T. Incidents Information Sharing Platform for Distributed Attack Detection. IEEE Open J. Commun. Soc. 2020, 1, 593–605.

[17].Chatzigiannakis, I.; Maiano, L.; Trakadas, P.; Anagnostopoulos, A.; Bacci, F.; Karkazis, P.; Spirakis, P.; Zahariadis, T. Data-Driven Intrusion Detection for

[18].Ambient Intelligence. In Ambient Intelligence, Lecture Notes in Computer Science; Chatzigiannakis, I., De Ruyter, B., Mavrommati, I., Eds.; Springer: Cham, Switzerland, 2019; Volume 11912.

[19].Lagutin, D.; Bellesini, F.; Bragatto, T.; Cavadenti, A.; Croce, V.; Kortesniemi, Y.; Leligou, H.C.; Oikonomidis, Y.; Polyzos, G.C.; Raveduto, G.; et al. Secure Open Federation of IoT Platforms Through Interledger Technologies-The SOFIE Approach. In Proceedings of the European Conference on Networks and Communications (EuCNC), Valencia, Spain, 18–21 June 2019; pp. 518–522.

[20].Lagutin, D.; Anton, P.; Bellesini, F.; Bragatto, T.; Cavadenti, A.; Croce, V.; Fotiou, N.; Haavala, M.; Kortesniemi, Y.; Leligou, H.C.; et al. The SOFIE Approach to Address the Security and Privacy of the IoT using Interledger Technologies. In Security and Privacy in Internet of Things: Challenges and Solutions; Ramos, J.L.H., Skarmeta, A., Eds.; IOS Press: Amsterdam, The Netherlands, 2020.

[21].Yao, X.; Zhou, J.; Zhang, J.; Boër, C.R. From Intelligent Manufacturing to Smart Manufacturing for Industry 4.0 Driven by Next Generation Artificial Intelligence and Further On. In Proceedings of the 5th International Conference on Enterprise Systems (ES), Beijing, China, 22–24 September 2017; pp. 311–318.

[22].Shin, K.; Park, H. Smart Manufacturing Systems Engineering for Designing Smart Product-Quality Monitoring System in the Industry 4.0. In Proceedings of the 19th International Conference on Control, Automation and Systems (ICCAS), Jeju, Korea, 15–18 October 2019; pp. 1693–1698.

[23].Bresniker, K.; Gavrilovska, A.; Holt, J.; Milojicic, D.; Tran, T. Grand Challenge: Applying Artificial Intelligence and Machine Learning to Cybersecurity. Computer 2019, 52, 45–52.

[24].Zeadally, S.; Adi, E.; Baig, Z.; Khan, I.A. Harnessing Artificial Intelligence Capabilities to Improve Cybersecurity. IEEE Access 2020, 8, 23817–23837.

[25].Tao, F.; Qi, Q.; Wang, L.; Nee, A.Y.C. Digital Twins and Cyber–Physical Systems toward Smart Manufacturing and Industry 4.0. Engineering 2019, 5, 653–661.

[26].Sittón-Candanedo, I.; Alonso, R.C.; Rodríguez-González, S.; Alberto García Coria, J.; De La Prieta, F. Edge Computing Architectures in Industry 4.0: A General Survey and Comparison. In International Workshop on Soft Computing Models in Industrial and Environmental Applications; Springer: Cham, Switzerland, 2019; Volume 950.

[27].Khan, M.; Wu, X.; Xu, X.; Dou, W. Big data challenges and opportunities in the hype of Industry 4.0. In Proceedings of the IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017.

[28].Zhong, R.Y.; Xu, X.; Klotz, E.; Newman, S.T. Intelligent manufacturing in the context of Industry 4.0: A Review.Engineering 2017, 3, 616–630.

[29].Zeid, A.; Sundaram, S.; Moghaddam, M.; Kamarthi, S.; Marion, T. Interoperability in Smart Manufacturing: Research Challenges. Machines 2019, 7, 21.

[30].Rosendahl, R.; Calá, A.; Kirchheim, K.; Lüder, A.; D’Agostino, N. Towards Smart Factory: Multi-Agent Integration on Industrial Standards for Service-oriented Communication and Semantic Data Exchange. WOA 2018, 124–132.

[31].Zhang, W.; Yang, D.; Wang, H. Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey. IEEE Syst. J. 2019, 13, 2213–2227.

[32].Chen, X.; Zhao, J. Research on the model and application of knowledge-based industrial design. In Proceedings of the International Technology and Innovation Conference (ITIC 2006), Hangzhou, China, 6–7 November 2006; pp. 1369–1374.

[33].Kattepur, A.; Dey, S.; Balamuralidhar, P. Knowledge Based Hierarchical Decomposition of Industry 4.0 Robotic Automation Tasks. In Proceedings of the IECON 2018–44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 21–23 October 2018; pp. 3665–3672.

[34].Ferrari, P.; Rinaldi, S.; Sisinni, E.; Colombo, F.; Ghelfi, F.; Maffei, D.; Malara, M. Performance evaluation of full-cloud and edge-cloud architectures for Industrial IoT anomaly detection based on deep learning. In Proceedings of the 2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0&IoT), Naples, Italy, 4–6 June 2019; pp. 420–425.

[35].Vretos, N.; Daras, P.; Asteriadis, S.; Hortal, E.; Ghaleb, E.; Spyrou, E.; Leligou, H.C.; Karkazis, P.; Trakadas, P.; Assimakopoulos, K. Exploiting sensing devices availability in AR/VR deployments to foster engagement. Virtual Real. 2019, 23, 399–410.

**CHAPTER -8**

**SAMPLE CODE**

**8.Sample code**

**Main.py**

import os

import time

import PySimpleGUI as sg

import industri\_temp as tempp

import industri\_heat as heatt

import industri\_gas as gas

symp = ['Yes ','No ']

import random

m = round(random.uniform(1.5, 10.5),1)

n = round(random.uniform(1.5, 10.5),1)

o = round(random.uniform(1.5, 10.5),1)

p = round(random.uniform(1.5, 10.5),1)

q = round(random.uniform(1.5, 10.5),1)

pollurants = [m,n,o,p,q]

ports = ["COM 1","COM 2","COM 5"]

layout = [

[sg.Text('SMART INDUSTRIAL AUTMOMATION', font='sfprodisplay 23 bold' ,pad =(440,50))],

[sg.Text('',pad =(30,18))],

[sg.Text('',pad =(30,18))],

[sg.Text('Select Sensor',pad =(0,8),font='sfprodisplay 15 bold ')],

[sg.Text(' ' , key='-OUTPUT-',pad =(0,8))],

[sg.Button('Temprature',key='predicT' , size=(25,2)),sg.Button('Pressure',key='predicH' , size=(25,2)),sg.Button('Gas',key='predicG' , size=(25,2))]

]

global window

window = sg.Window('ML PREDICTION Panel', size=(1920,1080) , location=(0,0),element\_justification='c', resizable=True,return\_keyboard\_events=True).Layout(layout)

#CONFIRM

while True:

event, values = window.Read()

if event == 'predicT':

tempp.starttemp()

if event == 'predicH':

heatt.startheat()

if event == 'predicG':

gas.startgas()

if event is None or event == 'exit':

window.close()

break

**industri\_temp.py**

import time

from threading import Thread

def sleeper(i,upvalue):

a = upvalue

window.FindElement('updated1'+i).Update(' '+str(a),text\_color='red')

while a != 0:

if a == 25:

window.FindElement('updated1'+i).Update(' '+str(a),text\_color='green')

window.FindElement('status\_'+i).Update(' Airconditioning-OFF ',text\_color='red')

window.FindElement('updated'+i).Update(' Stable',text\_color='green')

break

print (a)

time.sleep(1)

a = a-1

window.FindElement('updated1'+i).Update(' '+str(a),text\_color='red')

window.FindElement('status\_'+i).Update(' Airconditioning-ON ',text\_color='green')

window.FindElement('updated'+i).Update(' UnStable',text\_color='red')

import csv

#reading rows

with open('pol\_data.csv', 'r') as f:

reader = csv.reader(f)

your\_list = list(reader)

print('total dataset : = ' , len(your\_list)-1)

with open('pol\_data.csv', 'rU') as infile:

# read the file as a dictionary for each row ({header : value})

reader = csv.DictReader(infile)

data = {}

for row in reader:

for header, value in row.items():

try:

data[header].append(value)

except KeyError:

data[header] = [value]

with open('pol\_data.csv', 'rU') as infile:

# read the file as a dictionary for each row ({header : value})

reader = csv.DictReader(infile)

test = {}

for row in reader:

for header, value in row.items():

try:

test[header].append(value)

except KeyError:

test[header] = [value]

# extract the variables you want

At = data['SO2']

Ag = data['O3']

d1 = data['NO2']

d2 = data['PM10']

SF = data['PM 2.5']

import PySimpleGUI as sg

symp = ['Yes ','No ']

import random

m = round(random.uniform(1.5, 10.5),1)

n = round(random.uniform(1.5, 10.5),1)

o = round(random.uniform(1.5, 10.5),1)

p = round(random.uniform(1.5, 10.5),1)

q = round(random.uniform(1.5, 10.5),1)

pollurants = [m,n,o,p,q]

ports = ["COM 1","COM 2","COM 5"]

def starttemp():

layout = [

[sg.Text('SMART INDUSTRIAL AUTMOMATION', font='sfprodisplay 23 bold' ,pad =(440,50))],

[sg.Text('',pad =(30,8))],

[sg.Text('Sensor Values(Ref@25 oC)',pad =(30,8),font='sfprodisplay 15 bold'),sg.Text('Machine Enivironment',pad =(100,8),font='sfprodisplay 15 bold '),sg.Text('System Status',pad =(0,8),font='sfprodisplay 15 bold '),sg.Text(' Updated Sensor Value',pad =(0,8),font='sfprodisplay 15 bold ')],

[sg.Text('Time T1',font='sfprodisplay 15 bold'),sg.InputText(enable\_events=True, key='InputText\_1', size=(10,2)), sg.Button('OK', key='ok\_1'),sg.Text(' Airconditioning ', pad =(10,0),font='sfprodisplay 15 ',key='status\_1'),sg.Text(' Stable', font='sfprodisplay 15 bold', key='updated1'),sg.Text(' XX', font='sfprodisplay 15 bold', key='updated11')],

[sg.Text('Time T2',font='sfprodisplay 15 bold'),sg.InputText(enable\_events=True, key='InputText\_2', size=(10,2)), sg.Button('OK', key='ok\_2'),sg.Text(' Airconditioning ', pad =(10,0),font='sfprodisplay 15 ',key='status\_2'),sg.Text(' Stable', font='sfprodisplay 15 bold', key='updated2'),sg.Text(' XX', font='sfprodisplay 15 bold', key='updated12')],

[sg.Text('Time T3',font='sfprodisplay 15 bold'),sg.InputText(enable\_events=True, key='InputText\_3', size=(10,2)), sg.Button('OK', key='ok\_3'),sg.Text(' Airconditioning ', pad =(10,0),font='sfprodisplay 15 ',key='status\_3'),sg.Text(' Stable', font='sfprodisplay 15 bold', key='updated3'),sg.Text(' XX', font='sfprodisplay 15 bold', key='updated13')],

[sg.Text('Time T4',font='sfprodisplay 15 bold'),sg.InputText(enable\_events=True, key='InputText\_4', size=(10,2)), sg.Button('OK', key='ok\_4'),sg.Text(' Airconditioning ', pad =(10,0),font='sfprodisplay 15 ',key='status\_4'),sg.Text(' Stable', font='sfprodisplay 15 bold', key='updated4'),sg.Text(' XX', font='sfprodisplay 15 bold', key='updated14')],

[sg.Text('Time T5',font='sfprodisplay 15 bold'),sg.InputText(enable\_events=True, key='InputText\_5', size=(10,2)), sg.Button('OK', key='ok\_5'),sg.Text(' Airconditioning ', pad =(10,0),font='sfprodisplay 15 ',key='status\_5'),sg.Text(' Stable', font='sfprodisplay 15 bold', key='updated5'),sg.Text(' XX', font='sfprodisplay 15 bold', key='updated15')],

[sg.Text(' ' , key='-OUTPUT-',pad =(0,8))],

]

global window

window = sg.Window('ML PREDICTION Panel', size=(1920,1080) , location=(0,0),element\_justification='c', resizable=True,return\_keyboard\_events=True).Layout(layout)

#CONFIRM

while True:

event, values = window.Read()

if event == 'ok\_1':

if values['InputText\_1'] != "":

m = int(values['InputText\_1'])

if m<=25:

window.FindElement('status\_1').Update(' Airconditioning-OFF ',text\_color='red')

window.FindElement('updated1').Update(' Stable',text\_color='green')

else:

window.FindElement('status\_1').Update(' Airconditioning-ON ',text\_color='green')

window.FindElement('updated1').Update(' UnStable',text\_color='red')

# Create new threads

t = Thread(target=sleeper, args=('1',m))

t.start()

if event == 'ok\_2':

if values['InputText\_2'] != "":

m = int(values['InputText\_2'])

if m<=25:

window.FindElement('status\_2').Update(' Airconditioning ',text\_color='red')

window.FindElement('updated2').Update(' Stable',text\_color='green')

else:

window.FindElement('status\_2').Update(' Airconditioning ',text\_color='green')

window.FindElement('updated2').Update(' UnStable',text\_color='red')

if event == 'ok\_3':

if values['InputText\_3'] != "":

m = int(values['InputText\_3'])

if m<=25:

window.FindElement('status\_3').Update(' Airconditioning ',text\_color='red')

window.FindElement('updated3').Update(' Stable',text\_color='green')

else:

window.FindElement('status\_3').Update(' Airconditioning ',text\_color='green')

window.FindElement('updated3').Update(' UnStable',text\_color='red')

if event == 'ok\_4':

if values['InputText\_4'] != "":

m = int(values['InputText\_4'])

if m<=25:

window.FindElement('status\_4').Update(' Airconditioning ',text\_color='red')

window.FindElement('updated4').Update(' Stable',text\_color='green')

else:

window.FindElement('status\_4').Update(' Airconditioning ',text\_color='green')

window.FindElement('updated4').Update(' UnStable',text\_color='red')

if event == 'ok\_5':

if values['InputText\_5'] != "":

m = int(values['InputText\_5'])

if m<=25:

window.FindElement('status\_5').Update(' Airconditioning ',text\_color='red')

window.FindElement('updated5').Update(' Stable',text\_color='green')

else:

window.FindElement('status\_5').Update(' Airconditioning ',text\_color='green')

window.FindElement('updated5').Update(' UnStable',text\_color='red')

if event in ('\r', 'special 16777220', 'special 16777221'):

print(event, values)

print(values['Fever\_combo\_1'])

if int(values['Fever\_combo\_1']) <= 10 and int(values['Fever\_combo\_1']) >= 1:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('OFF',text\_color='red')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

while True:

print(values['Fever\_combo\_2'])

event, values = window.Read()

if event in ('\r', 'special 16777220', 'special 16777221'):

if int(values['Fever\_combo\_2']) <= 20 and int(values['Fever\_combo\_2']) > 10:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

else:

if int(values['Fever\_combo\_2']) <= 10 and int(values['Fever\_combo\_2']) >= 1:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('OFF',text\_color='red')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

while True:

print(values['Fever\_combo\_3'])

event, values = window.Read()

if event in ('\r', 'special 16777220', 'special 16777221'):

if int(values['Fever\_combo\_3']) <= 30 and int(values['Fever\_combo\_3']) > 20:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('ON',text\_color='green')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_3']) <= 10 and int(values['Fever\_combo\_3']) >= 1:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('OFF',text\_color='red')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_3']) <= 20 and int(values['Fever\_combo\_3']) >= 10:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

while True:

print(values['Fever\_combo\_4'])

event, values = window.Read()

if event in ('\r', 'special 16777220', 'special 16777221'):

if int(values['Fever\_combo\_4']) <= 40 and int(values['Fever\_combo\_4']) > 30:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('ON',text\_color='green')

window.FindElement('status4').Update('ON',text\_color='green')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_4']) <= 10 and int(values['Fever\_combo\_4']) >= 1:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('OFF',text\_color='red')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_4']) <= 20 and int(values['Fever\_combo\_4']) >= 10:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_4']) <= 30 and int(values['Fever\_combo\_4']) >= 20:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('ON',text\_color='green')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

while True:

print(values['Fever\_combo\_5'])

event, values = window.Read()

if event in ('\r', 'special 16777220', 'special 16777221'):

if int(values['Fever\_combo\_5']) <= 50 and int(values['Fever\_combo\_5']) > 40:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('ON',text\_color='green')

window.FindElement('status4').Update('ON',text\_color='green')

window.FindElement('status5').Update('ON',text\_color='green')

break

elif int(values['Fever\_combo\_5']) <= 10 and int(values['Fever\_combo\_5']) >= 1:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('OFF',text\_color='red')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_5']) <= 20 and int(values['Fever\_combo\_5']) >= 10:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_5']) <= 30 and int(values['Fever\_combo\_5']) >= 20:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('ON',text\_color='green')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_5']) <= 40 and int(values['Fever\_combo\_5']) >= 30:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('ON',text\_color='green')

window.FindElement('status4').Update('ON',text\_color='green')

window.FindElement('status5').Update('OFF',text\_color='red')

break

'''

if event is None or event == 'exit':

window.close()

break

**industri\_heat.py**

import time

from threading import Thread

def sleeper(i,upvalue):

a = upvalue

window.FindElement('updated1'+i).Update(' '+str(a),text\_color='red')

while a != 0:

if a == 10:

window.FindElement('updated1'+i).Update(' '+str(a),text\_color='green')

window.FindElement('status\_'+i).Update(' Valve-OFF ',text\_color='red')

window.FindElement('updated'+i).Update(' Stable',text\_color='green')

break

print (a)

time.sleep(1)

a = a-1

window.FindElement('updated1'+i).Update(' '+str(a),text\_color='red')

window.FindElement('status\_'+i).Update(' Valve-ON ',text\_color='green')

window.FindElement('updated'+i).Update(' UnStable',text\_color='red')

import csv

#reading rows

with open('pol\_data.csv', 'r') as f:

reader = csv.reader(f)

your\_list = list(reader)

print('total dataset : = ' , len(your\_list)-1)

with open('pol\_data.csv', 'rU') as infile:

# read the file as a dictionary for each row ({header : value})

reader = csv.DictReader(infile)

data = {}

for row in reader:

for header, value in row.items():

try:

data[header].append(value)

except KeyError:

data[header] = [value]

with open('pol\_data.csv', 'rU') as infile:

# read the file as a dictionary for each row ({header : value})

reader = csv.DictReader(infile)

test = {}

for row in reader:

for header, value in row.items():

try:

test[header].append(value)

except KeyError:

test[header] = [value]

# extract the variables you want

At = data['SO2']

Ag = data['O3']

d1 = data['NO2']

d2 = data['PM10']

SF = data['PM 2.5']

import PySimpleGUI as sg

symp = ['Yes ','No ']

import random

m = round(random.uniform(1.5, 10.5),1)

n = round(random.uniform(1.5, 10.5),1)

o = round(random.uniform(1.5, 10.5),1)

p = round(random.uniform(1.5, 10.5),1)

q = round(random.uniform(1.5, 10.5),1)

pollurants = [m,n,o,p,q]

ports = ["COM 1","COM 2","COM 5"]

def startheat():

layout = [

[sg.Text('SMART INDUSTRIAL AUTMOMATION', font='sfprodisplay 23 bold' ,pad =(440,50))],

[sg.Text('',pad =(30,8))],

[sg.Text('Sensor Values(Ref@10 pascals)',pad =(30,5),font='sfprodisplay 15 bold'),sg.Text('Machine Enivironment',pad =(100,5),font='sfprodisplay 15 bold '),sg.Text('System Status',pad =(0,8),font='sfprodisplay 15 bold '),sg.Text(' Updated Sensor Value',pad =(0,8),font='sfprodisplay 15 bold ')],

[sg.Text('Time T1',font='sfprodisplay 15 bold'),sg.InputText(enable\_events=True, key='InputText\_1', size=(10,2)), sg.Button('OK', key='ok\_1'),sg.Text(' Valve ', pad =(10,0),font='sfprodisplay 15 ',key='status\_1'),sg.Text(' Stable', font='sfprodisplay 15 bold', key='updated1'),sg.Text(' XX', font='sfprodisplay 15 bold', key='updated11')],

[sg.Text('Time T2',font='sfprodisplay 15 bold'),sg.InputText(enable\_events=True, key='InputText\_2', size=(10,2)), sg.Button('OK', key='ok\_2'),sg.Text(' Valve ', pad =(10,0),font='sfprodisplay 15 ',key='status\_2'),sg.Text(' Stable', font='sfprodisplay 15 bold', key='updated2'),sg.Text(' XX', font='sfprodisplay 15 bold', key='updated12')],

[sg.Text('Time T3',font='sfprodisplay 15 bold'),sg.InputText(enable\_events=True, key='InputText\_3', size=(10,2)), sg.Button('OK', key='ok\_3'),sg.Text(' Valve ', pad =(10,0),font='sfprodisplay 15 ',key='status\_3'),sg.Text(' Stable', font='sfprodisplay 15 bold', key='updated3'),sg.Text(' XX', font='sfprodisplay 15 bold', key='updated13')],

[sg.Text('Time T4',font='sfprodisplay 15 bold'),sg.InputText(enable\_events=True, key='InputText\_4', size=(10,2)), sg.Button('OK', key='ok\_4'),sg.Text(' Valve ', pad =(10,0),font='sfprodisplay 15 ',key='status\_4'),sg.Text(' Stable', font='sfprodisplay 15 bold', key='updated4'),sg.Text(' XX', font='sfprodisplay 15 bold', key='updated14')],

[sg.Text('Time T5',font='sfprodisplay 15 bold'),sg.InputText(enable\_events=True, key='InputText\_5', size=(10,2)), sg.Button('OK', key='ok\_5'),sg.Text(' Valve ', pad =(10,0),font='sfprodisplay 15 ',key='status\_5'),sg.Text(' Stable', font='sfprodisplay 15 bold', key='updated5'),sg.Text(' XX', font='sfprodisplay 15 bold', key='updated15')],

[sg.Text(' ' , key='-OUTPUT-',pad =(0,8))],

]

global window

window = sg.Window('ML PREDICTION Panel', size=(1920,1080) , location=(0,0),element\_justification='c', resizable=True,return\_keyboard\_events=True).Layout(layout)

#CONFIRM

while True:

event, values = window.Read()

if event == 'ok\_1':

if values['InputText\_1'] != "":

m = int(values['InputText\_1'])

if m<=10:

window.FindElement('status\_1').Update(' Valve-OFF ',text\_color='red')

window.FindElement('updated1').Update(' Stable',text\_color='green')

else:

window.FindElement('status\_1').Update(' Valve-ON ',text\_color='green')

window.FindElement('updated1').Update(' UnStable',text\_color='red')

# Create new threads

t = Thread(target=sleeper, args=('1',m))

t.start()

if event == 'ok\_2':

if values['InputText\_2'] != "":

m = int(values['InputText\_2'])

if m<=25:

window.FindElement('status\_2').Update(' Airconditioning ',text\_color='red')

window.FindElement('updated2').Update(' Stable',text\_color='green')

else:

window.FindElement('status\_2').Update(' Airconditioning ',text\_color='green')

window.FindElement('updated2').Update(' UnStable',text\_color='red')

if event == 'ok\_3':

if values['InputText\_3'] != "":

m = int(values['InputText\_3'])

if m<=25:

window.FindElement('status\_3').Update(' Airconditioning ',text\_color='red')

window.FindElement('updated3').Update(' Stable',text\_color='green')

else:

window.FindElement('status\_3').Update(' Airconditioning ',text\_color='green')

window.FindElement('updated3').Update(' UnStable',text\_color='red')

if event == 'ok\_4':

if values['InputText\_4'] != "":

m = int(values['InputText\_4'])

if m<=25:

window.FindElement('status\_4').Update(' Airconditioning ',text\_color='red')

window.FindElement('updated4').Update(' Stable',text\_color='green')

else:

window.FindElement('status\_4').Update(' Airconditioning ',text\_color='green')

window.FindElement('updated4').Update(' UnStable',text\_color='red')

if event == 'ok\_5':

if values['InputText\_5'] != "":

m = int(values['InputText\_5'])

if m<=25:

window.FindElement('status\_5').Update(' Airconditioning ',text\_color='red')

window.FindElement('updated5').Update(' Stable',text\_color='green')

else:

window.FindElement('status\_5').Update(' Airconditioning ',text\_color='green')

window.FindElement('updated5').Update(' UnStable',text\_color='red')

'''

if event in ('\r', 'special 16777220', 'special 16777221'):

print(event, values)

print(values['Fever\_combo\_1'])

if int(values['Fever\_combo\_1']) <= 10 and int(values['Fever\_combo\_1']) >= 1:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('OFF',text\_color='red')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

while True:

print(values['Fever\_combo\_2'])

event, values = window.Read()

if event in ('\r', 'special 16777220', 'special 16777221'):

if int(values['Fever\_combo\_2']) <= 20 and int(values['Fever\_combo\_2']) > 10:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

else:

if int(values['Fever\_combo\_2']) <= 10 and int(values['Fever\_combo\_2']) >= 1:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('OFF',text\_color='red')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

while True:

print(values['Fever\_combo\_3'])

event, values = window.Read()

if event in ('\r', 'special 16777220', 'special 16777221'):

if int(values['Fever\_combo\_3']) <= 30 and int(values['Fever\_combo\_3']) > 20:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('ON',text\_color='green')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_3']) <= 10 and int(values['Fever\_combo\_3']) >= 1:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('OFF',text\_color='red')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_3']) <= 20 and int(values['Fever\_combo\_3']) >= 10:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

while True:

print(values['Fever\_combo\_4'])

event, values = window.Read()

if event in ('\r', 'special 16777220', 'special 16777221'):

if int(values['Fever\_combo\_4']) <= 40 and int(values['Fever\_combo\_4']) > 30:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('ON',text\_color='green')

window.FindElement('status4').Update('ON',text\_color='green')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_4']) <= 10 and int(values['Fever\_combo\_4']) >= 1:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('OFF',text\_color='red')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_4']) <= 20 and int(values['Fever\_combo\_4']) >= 10:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_4']) <= 30 and int(values['Fever\_combo\_4']) >= 20:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('ON',text\_color='green')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

while True:

print(values['Fever\_combo\_5'])

event, values = window.Read()

if event in ('\r', 'special 16777220', 'special 16777221'):

if int(values['Fever\_combo\_5']) <= 50 and int(values['Fever\_combo\_5']) > 40:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('ON',text\_color='green')

window.FindElement('status4').Update('ON',text\_color='green')

window.FindElement('status5').Update('ON',text\_color='green')

break

elif int(values['Fever\_combo\_5']) <= 10 and int(values['Fever\_combo\_5']) >= 1:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('OFF',text\_color='red')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_5']) <= 20 and int(values['Fever\_combo\_5']) >= 10:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_5']) <= 30 and int(values['Fever\_combo\_5']) >= 20:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('ON',text\_color='green')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_5']) <= 40 and int(values['Fever\_combo\_5']) >= 30:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('ON',text\_color='green')

window.FindElement('status4').Update('ON',text\_color='green')

window.FindElement('status5').Update('OFF',text\_color='red')

break

'''

if event is None or event == 'exit':

window.close()

break

#startheat()

Industri.py

import time

from threading import Thread

def sleeper(i,upvalue):

a = upvalue

window.FindElement('updated1'+i).Update(' '+str(a),text\_color='red')

while a != 0:

if a == 25:

window.FindElement('updated1'+i).Update(' '+str(a),text\_color='green')

window.FindElement('status\_'+i).Update(' Airconditioning-OFF ',text\_color='red')

window.FindElement('updated'+i).Update(' Stable',text\_color='green')

break

print (a)

time.sleep(1)

a = a-1

window.FindElement('updated1'+i).Update(' '+str(a),text\_color='red')

window.FindElement('status\_'+i).Update(' Airconditioning-ON ',text\_color='green')

window.FindElement('updated'+i).Update(' UnStable',text\_color='red')

import csv

#reading rows

with open('pol\_data.csv', 'r') as f:

reader = csv.reader(f)

your\_list = list(reader)

print('total dataset : = ' , len(your\_list)-1)

with open('pol\_data.csv', 'rU') as infile:

# read the file as a dictionary for each row ({header : value})

reader = csv.DictReader(infile)

data = {}

for row in reader:

for header, value in row.items():

try:

data[header].append(value)

except KeyError:

data[header] = [value]

with open('pol\_data.csv', 'rU') as infile:

# read the file as a dictionary for each row ({header : value})

reader = csv.DictReader(infile)

test = {}

for row in reader:

for header, value in row.items():

try:

test[header].append(value)

except KeyError:

test[header] = [value]

# extract the variables you want

At = data['SO2']

Ag = data['O3']

d1 = data['NO2']

d2 = data['PM10']

SF = data['PM 2.5']

import PySimpleGUI as sg

symp = ['Yes ','No ']

import random

m = round(random.uniform(1.5, 10.5),1)

n = round(random.uniform(1.5, 10.5),1)

o = round(random.uniform(1.5, 10.5),1)

p = round(random.uniform(1.5, 10.5),1)

q = round(random.uniform(1.5, 10.5),1)

pollurants = [m,n,o,p,q]

ports = ["COM 1","COM 2","COM 5"]

layout = [

[sg.Text('SMART INDUSTRIAL AUTMOMATION', font='sfprodisplay 23 bold' ,pad =(440,50))],

[sg.Text('',pad =(30,8))],

[sg.Text('Sensor Values(Ref@25 oC)',pad =(30,8),font='sfprodisplay 15 bold'),sg.Text('Machine Enivironment',pad =(100,8),font='sfprodisplay 15 bold '),sg.Text('System Status',pad =(0,8),font='sfprodisplay 15 bold '),sg.Text(' Updated Sensor Value',pad =(0,8),font='sfprodisplay 15 bold ')],

[sg.Text('Time T1',font='sfprodisplay 15 bold'),sg.InputText(enable\_events=True, key='InputText\_1', size=(10,2)), sg.Button('OK', key='ok\_1'),sg.Text(' Airconditioning ', pad =(10,0),font='sfprodisplay 15 ',key='status\_1'),sg.Text(' Stable', font='sfprodisplay 15 bold', key='updated1'),sg.Text(' XX', font='sfprodisplay 15 bold', key='updated11')],

[sg.Text('Time T2',font='sfprodisplay 15 bold'),sg.InputText(enable\_events=True, key='InputText\_2', size=(10,2)), sg.Button('OK', key='ok\_2'),sg.Text(' Airconditioning ', pad =(10,0),font='sfprodisplay 15 ',key='status\_2'),sg.Text(' Stable', font='sfprodisplay 15 bold', key='updated2'),sg.Text(' XX', font='sfprodisplay 15 bold', key='updated12')],

[sg.Text('Time T3',font='sfprodisplay 15 bold'),sg.InputText(enable\_events=True, key='InputText\_3', size=(10,2)), sg.Button('OK', key='ok\_3'),sg.Text(' Airconditioning ', pad =(10,0),font='sfprodisplay 15 ',key='status\_3'),sg.Text(' Stable', font='sfprodisplay 15 bold', key='updated3'),sg.Text(' XX', font='sfprodisplay 15 bold', key='updated13')],

[sg.Text('Time T4',font='sfprodisplay 15 bold'),sg.InputText(enable\_events=True, key='InputText\_4', size=(10,2)), sg.Button('OK', key='ok\_4'),sg.Text(' Airconditioning ', pad =(10,0),font='sfprodisplay 15 ',key='status\_4'),sg.Text(' Stable', font='sfprodisplay 15 bold', key='updated4'),sg.Text(' XX', font='sfprodisplay 15 bold', key='updated14')],

[sg.Text('Time T5',font='sfprodisplay 15 bold'),sg.InputText(enable\_events=True, key='InputText\_5', size=(10,2)), sg.Button('OK', key='ok\_5'),sg.Text(' Airconditioning ', pad =(10,0),font='sfprodisplay 15 ',key='status\_5'),sg.Text(' Stable', font='sfprodisplay 15 bold', key='updated5'),sg.Text(' XX', font='sfprodisplay 15 bold', key='updated15')],

[sg.Text(' ' , key='-OUTPUT-',pad =(0,8))],

]

global window

window = sg.Window('ML PREDICTION Panel', size=(1920,1080) , location=(0,0),element\_justification='c', resizable=True,return\_keyboard\_events=True).Layout(layout)

#CONFIRM

while True:

event, values = window.Read()

if event == 'ok\_1':

if values['InputText\_1'] != "":

m = int(values['InputText\_1'])

if m<=25:

window.FindElement('status\_1').Update(' Airconditioning-OFF ',text\_color='red')

window.FindElement('updated1').Update(' Stable',text\_color='green')

else:

window.FindElement('status\_1').Update(' Airconditioning-ON ',text\_color='green')

window.FindElement('updated1').Update(' UnStable',text\_color='red')

# Create new threads

t = Thread(target=sleeper, args=('1',m))

t.start()

if event == 'ok\_2':

if values['InputText\_2'] != "":

m = int(values['InputText\_2'])

if m<=25:

window.FindElement('status\_2').Update(' Airconditioning ',text\_color='red')

window.FindElement('updated2').Update(' Stable',text\_color='green')

else:

window.FindElement('status\_2').Update(' Airconditioning ',text\_color='green')

window.FindElement('updated2').Update(' UnStable',text\_color='red')

if event == 'ok\_3':

if values['InputText\_3'] != "":

m = int(values['InputText\_3'])

if m<=25:

window.FindElement('status\_3').Update(' Airconditioning ',text\_color='red')

window.FindElement('updated3').Update(' Stable',text\_color='green')

else:

window.FindElement('status\_3').Update(' Airconditioning ',text\_color='green')

window.FindElement('updated3').Update(' UnStable',text\_color='red')

if event == 'ok\_4':

if values['InputText\_4'] != "":

m = int(values['InputText\_4'])

if m<=25:

window.FindElement('status\_4').Update(' Airconditioning ',text\_color='red')

window.FindElement('updated4').Update(' Stable',text\_color='green')

else:

window.FindElement('status\_4').Update(' Airconditioning ',text\_color='green')

window.FindElement('updated4').Update(' UnStable',text\_color='red')

if event == 'ok\_5':

if values['InputText\_5'] != "":

m = int(values['InputText\_5'])

if m<=25:

window.FindElement('status\_5').Update(' Airconditioning ',text\_color='red')

window.FindElement('updated5').Update(' Stable',text\_color='green')

else:

window.FindElement('status\_5').Update(' Airconditioning ',text\_color='green')

window.FindElement('updated5').Update(' UnStable',text\_color='red')

'''

if event in ('\r', 'special 16777220', 'special 16777221'):

print(event, values)

print(values['Fever\_combo\_1'])

if int(values['Fever\_combo\_1']) <= 10 and int(values['Fever\_combo\_1']) >= 1:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('OFF',text\_color='red')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

while True:

print(values['Fever\_combo\_2'])

event, values = window.Read()

if event in ('\r', 'special 16777220', 'special 16777221'):

if int(values['Fever\_combo\_2']) <= 20 and int(values['Fever\_combo\_2']) > 10:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

else:

if int(values['Fever\_combo\_2']) <= 10 and int(values['Fever\_combo\_2']) >= 1:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('OFF',text\_color='red')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

while True:

print(values['Fever\_combo\_3'])

event, values = window.Read()

if event in ('\r', 'special 16777220', 'special 16777221'):

if int(values['Fever\_combo\_3']) <= 30 and int(values['Fever\_combo\_3']) > 20:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('ON',text\_color='green')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_3']) <= 10 and int(values['Fever\_combo\_3']) >= 1:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('OFF',text\_color='red')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_3']) <= 20 and int(values['Fever\_combo\_3']) >= 10:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

while True:

print(values['Fever\_combo\_4'])

event, values = window.Read()

if event in ('\r', 'special 16777220', 'special 16777221'):

if int(values['Fever\_combo\_4']) <= 40 and int(values['Fever\_combo\_4']) > 30:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('ON',text\_color='green')

window.FindElement('status4').Update('ON',text\_color='green')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_4']) <= 10 and int(values['Fever\_combo\_4']) >= 1:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('OFF',text\_color='red')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_4']) <= 20 and int(values['Fever\_combo\_4']) >= 10:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_4']) <= 30 and int(values['Fever\_combo\_4']) >= 20:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('ON',text\_color='green')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

while True:

print(values['Fever\_combo\_5'])

event, values = window.Read()

if event in ('\r', 'special 16777220', 'special 16777221'):

if int(values['Fever\_combo\_5']) <= 50 and int(values['Fever\_combo\_5']) > 40:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('ON',text\_color='green')

window.FindElement('status4').Update('ON',text\_color='green')

window.FindElement('status5').Update('ON',text\_color='green')

break

elif int(values['Fever\_combo\_5']) <= 10 and int(values['Fever\_combo\_5']) >= 1:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('OFF',text\_color='red')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_5']) <= 20 and int(values['Fever\_combo\_5']) >= 10:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('OFF',text\_color='red')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_5']) <= 30 and int(values['Fever\_combo\_5']) >= 20:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('ON',text\_color='green')

window.FindElement('status4').Update('OFF',text\_color='red')

window.FindElement('status5').Update('OFF',text\_color='red')

break

elif int(values['Fever\_combo\_5']) <= 40 and int(values['Fever\_combo\_5']) >= 30:

window.FindElement('status1').Update('ON',text\_color='green')

window.FindElement('status2').Update('ON',text\_color='green')

window.FindElement('status3').Update('ON',text\_color='green')

window.FindElement('status4').Update('ON',text\_color='green')

window.FindElement('status5').Update('OFF',text\_color='red')

break

'''

if event is None or event == 'exit':

window.close()

break

**APPENDIX**



