HUMAN IMMUNE SYSTEM-INSPIRED FRAMEWORK FOR MANUFACTURING PROCESS DISRUPTION HANDLING

Zain Abbas[*], Sher Shah[**], Zubair Ahmad Khan[*], and Hamza Ahmad Khan[*]

* Department of Mechatronics Engineering University of Engineering & Technology Peshawar, Pakistan.

**Industrial Management and Innovation, Uppsala University, Sweden

Abstract-
Disruptions in the process industry often cause longer downtime and less effective operations. In the case of automated operations, when mistakes have the potential to significantly affect product quality and cost, this is especially important to keep in mind. Human immune system research has been used as a starting point for solutions to this problem. An AIS can be created to identify and segregate problems in the manufacturing process in the same way that our B and T cells identify and eliminate infectious microbes in the human body. Protégé was used to create an ontology focused on the immune system. The AIS model is preprogrammed to identify interruptions and launch an immediate countermeasure. The model can dynamically adapt its response to the unique interruption it meets by allocating weights to different criteria. Fifteen input/output factors affecting process downtime, system efficiency, and consumer-centric value were evaluated to put this approach to the test. They tested how well the AIS concept worked in a lab setting. As a result, this method offers a potentially fruitful way to enhance the effectiveness and dependability of automated operations in the process industry.

Index Terms- *Human immune system, Alarm trip point, disruption, Smart manufacturing system.*

I. INTRODUCTION

Due to the increasing complexity of modern production systems, they are more vulnerable than ever before to a wide range of interruptions. Process downtime, material loss, efficiency, and production are all negatively impacted by these errors and interruptions. The failure of a drive or a sensor to calibrate properly, the destruction of a tool, the loss of a gadget, etc., all fall into this category. Using an intelligent system is necessary in order to counteract the aforementioned disturbances that have a direct impact on the cost of the final product [1]. Even worse, if not dealt with properly, these interruptions can set off a chain reaction that will cause the entire manufacturing facility to crash and burn [2].

The human immune system provides a valuable model for creating effective systems that can manage and mitigate disruptions. By studying how the immune system functions, researchers can develop frameworks that can anticipate and resolve unexpected disturbances [3]. Despite the abundance of literature on the conceptual framework of fault detection and the incorporation of immune-based mechanisms into automated processes, there is still much room for improvement, and generic principles and procedures need further refinement. Although these methods have been shown to reduce the impact of interruptions, their implementation in a real-world system has been limited. The current work is an attempt to apply the concept of the HIS inspired Framework for Disruption Handling in manufacturing process (HISFDH), which was inspired by the human immune system, to a practical manufacturing environment. With the presented paradigm, rapid disruption assessments and automatic reaction production are both possible. Assigning weights relative to the database's predefined states allows for a real-time assessment of interruption impacts. The work is divided into the following sections: (1) an introduction; (2) a survey of HIS; (3) research on FDI; (4) a proposed HISFDH model; (5) a case study employing the model; and (6) a discussion of the findings and some suggestions for the future.

II. REVIEW OF HUMAN IMMUNE SYSTEM

The immune system protects the body from harmful substances called pathogens that enter the body through the skin. These pathogens disrupt normal bodily processes and so cause sickness. The HIS is further subdivided into the innate immune system and the adaptive immune system; both of these systems accomplish the same goals. Pathogens are stopped at the innate immune system's "gates," which include the skin, mucous membranes, gastric juice, epithelial cells, and so on. When pathogens enter the body, a second type of innate immunity kicks into gear; these cells consume the invaders via phagocytosis and display a portion of the pathogen's protein on their surface as a tag, earning them the name Antigen antigen-presenting cells (APC) [4]. Antigen-presenting cells allow the innate immune system to recognize a large variety of infections (APCs). To determine who or what is "Self" and who or what is "non-self," APCs use specific receptors for pathogens called Pathogens Receptor Cells (PRRs) on the surface of APCs [5].

The adaptive (acquired) immune response refers to the body's delayed response to all foreign diseases (see Figure 1). T-Cells and B-cells are the two types of cells in this immune system (Lymphocytes). T-cells can function as either a helper cell, promoting the generation and activation of B-cells (cd4), or as a killer cell, killing target cells (cd8). There are billions of B-cells, each containing a protein called B cells receptors (BCH) that interacts with a pathogen via an epitope. When a pathogen connects to a B cell, the cell goes through Receptor-mediated endocytosis to engulf and kill the intruder. The B cell then adheres to its surface a specific protein strand (a tag). These cells are also known as antigen-presentation cells (APCs) because of their role in presenting the MHC (Major Histocompatibility Complex) protein (Antigen Presenting Cells). APCs produce interleukin, which serves to anchor T lymphocytes. In response to interleukin, B cells undergo rapid mitotic divisions to tackle a large number of similar infections. During this process, some B and T cells mature into B and T memory cells that can recognize the removed pathogen. It lingers for a while to ensure a quick immunological response that keeps the body safe [6].

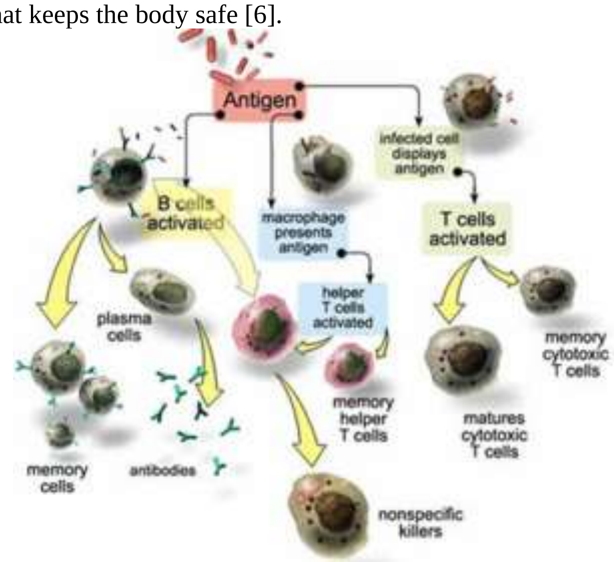

Figure 1: Human Immune System

The success of the human immune system has prompted the development of synthetic immune systems (AIS). Many scientists have attempted to replicate AIS's success by developing their own systems with similar capabilities, such as the ability to recognize antigens, to remember information, to rearrange itself, to frame immunological reactions, and to recognize patterns. Furthermore, immune-inspired systems possess characteristics such as reaction framing, generalization capability, and multilayer framing, which have been studied extensively [7]. The "Self and Non-Self" concept has been introduced in various fields, such as network security, milling procedures [8], fault detection and diagnosis [9], and network intrusion in computer algorithms [10]. In addressing production, planning, and scheduling issues in industrial settings, researchers have proposed various strategies, including immunological tactics [11]. In discussing anomaly detection and its applications in business [12].

III. LITERATURE REVIEW

There are three main aspects to this literature review: foreign direct investment models, alarm management strategies, and artificial immune approaches.

A. Models for the Detection and Diagnosis of Defects in Manufacturing Facilities

Fault detection and identification (FDI) is a typical strategy for dealing with disruptions. Two competing diagnostic models are proposed, with specifics for both continuous and logical fault-handling approaches [13]. A fault tree analysis (a more advanced logical and continuous diagnosis model) was devised to generate mayhem in manufacturing [14]. It proposes a method for process industry disruption based on the usage of hamming distance-dependent technology [15]. A more recent work [16] presents a system that may instantly generate a knowledge-based solution using PLC setup and circuit design. A technique suggested in [17] generates automatic PLC program code using a discrete event approach model for problem identification. Another aspect is that it simulates control logic using non-deterministic output automata [18]. [19] presents approaches based on minimum manufacturing process patterns that also leverage DESs for fault localization. When it comes to modeling control processes, [20] implements research based on autonomous automata. The finite state machine (FSM) technique for disruption handling [21] represents a significant improvement in this field. In the process industry, an amazing function-based analysis was proposed to determine the source of the sensor and actuator disagreement [22]. The Fault and Behavioral Anomaly Detection Tool for PLC Controlled Manufacturing Plant (FBMTP) offered here can detect anomalies in the process sector. The primary goal of this research is to use a novel and timed response mechanism to add real state transition time into existing non-deterministic models for fault rectification. A study in [23] provided a useful forward-backward algorithm-based approach for predicting remaining usable life and assessing online degradation. They created a model that connects the rate of degradation to its timeline. [24] created a concept that comprises developing an architecture for processing sensor data and providing fault-free zones in which machines can operate. A study in [25] uses modeled nonlinear observers to detect and locate process problems before activating the system's dynamic control logic to mitigate their effects. A study published in [26] looked at the costs of machine deterioration and offered many preventative maintenance measures to offset the higher costs. They also investigated the relationship between machine failure, quality measures, and product defect rate. According to a study published in [27], assembly line balance and component feeding increased the impacts of mixed integer models on consumer-centric value, lowering hazards in manufacturing facilities.

B. Limitations

In a nutshell, FDI approaches provide a unified nominal control process model as a means of dealing with specific disturbances across the production line. Most of these accomplish their aims by utilizing comparison approaches based on log data files of control indicators. It is nearly impossible to retain an accurate log of signal occurrence data in regulated systems with rapid data flow. However, to lengthen the scanning period, such devices require duplicate controllers. That makes it more challenging to achieve instantaneous responses in real-time.

C. Alarm Management

Whenever time a plant sounds an alarm, it needs prompt treatment. In the case of alarm, an ontology is defined that can be of use when analyzing a plant. The ontology-based linkage of alarms presented by [28] makes use of managed data. To fully realize the system's potential, it should be expanded to include more sophisticated monitoring capabilities. A filtering-based model for recommending stocks is shown in SIGARA [29]. However, it doesn't provide information on required tasks that need modeling in context. As stated in [30] a general ontology for representing context is offered alongside context-aware rules. It aims to demonstrate a framework that can be applied in numerous contexts.

D. Artificial Immune System

Artificial immune systems draw inspiration from natural immune systems as both are responsible for defending their respective hosts from harmful microorganisms. The objective of scientific research is to create artificial immune systems that possess superior capabilities such as identifying unique features of antigens, memorizing patterns, arranging memory, converting capabilities, framing immune reactions, learning from examples, processing information simultaneously, utilizing a multilayer framework, and exhibiting generalization abilities. AIS has numerous applications such as anomaly detection, fault diagnosis, computer and industrial network security, robotics, control, and optimization, among others. According to a study by [26], AIS can be used for both constant and combinatorial purposes. The immune system, as well as the concepts of negative selection and clonal selection, are crucial for the most common types of AIS. There was a presentation of a negative-selection artificial intelligence system by [31]. The use of immune system concepts and mechanisms in engineering has shown potential for various applications such as network intrusion detection, fault detection and diagnosis, and DNA computing. One of the methods used is clonal selection, where only cells possessing specific characteristics identified by antigen receptors will proliferate. The clonal selection algorithm has been applied to solve computational issues, scheduling, and resource allocation optimization. The artificial immune recognition system (AIRS) is another implementation of the clonal selection method.

Researchers have also developed frameworks and methodological guidelines for handling disruptions in production systems using immune-based approaches. However, there is a lack of focus on modeling disruption tactics, and there is a need for more attention and study in this area. Ontologies have been developed for dealing with disruptions in production systems, but more research is necessary to fully realize the potential of immune-based approaches in manufacturing. The proposed method of developing an immune-based ontology for the factory floor and implementing an immune-based disruption handling strategy in a functioning system addresses gaps in the existing literature. It is essential to give disruption handling models the attention they deserve in the software and domain of production systems to reduce process downtime and optimize efficiency, associated with each manufacturing process. The cell agents then use this information to make decisions on how to respond to disruptions in real time. One of the key advantages of using immune-based models in disruption management is their ability to adapt and learn from new disruptions. As the immune system encounters new pathogens, it adapts and develops new antibodies to defend against them. Similarly, in the manufacturing setting, an immune-based system can learn from new disruptions and use this knowledge to improve its response to future disruptions. This adaptability is particularly important in the manufacturing setting, where disruptions can have significant financial and logistical consequences. In conclusion, the use of immune-based models in disruption management has significant potential for improving efficiency and reducing downtime in manufacturing systems. By leveraging the principles of the immune system, these models can adapt and learn from new disruptions, leading to more effective and efficient responses. However, further research is needed to develop and refine these models for use in specific manufacturing settings.

IV. RESEARCH METHODOLOGY

For clarity, this paper suggests a two-step process.

1. The creation of an immunological ontology
2. Progress in immune-based disruption handling

We divide the process into two groups right away: physical and digital inputs/tags. Then, based on the ability to detect digital or physical labels, ontologies are created, and various agents are built. The electrical inputs and outputs, such as drives, valves, actuators, and so on, are labeled. Although virtual tags have no physical existence, they can be quite useful in automating the method for the coder. Virtual tags represent internal alarms, fault bits, reaction bits, analog sensor ranges, and so forth.

The second level of methodology is a mimic of the immune system's function in real-world organisms. Foreign pathogens can be detected by human neutrophils and macrophage cells. In response to a foreign pathogen, T cells generate interleukin1, 2, and 3 signals, which cause B cells to multiply exponentially, producing enough antibodies to load the threat (pathogen) and render it harmless. Cell agents might also be used to represent a manufacturing plant to ensure that it receives consistent information from the control architecture. Following immune system logic, both actual (i.e. Antigen) and fake (i.e. APC) data are compared to the baseline (i.e. Normal Process) (i.e. normal body cell). We are continually comparing and evaluating the acquired data with each scan. When an uncommon incidence is discovered, the appropriate B and T agents are notified. The correct response is achieved as a result of the antibodies generated. The database contains a collection of reactions that are defined by a certain occurrence. The introduction of this strategy boosts the immune system considerably (already in literature). We provide a Fault Tolerant agent that continuously evaluates the anomaly (with each scan). The fault-tolerant agent intelligently selects the production flow based on the weights assigned to the problem bits, assisting in the reduction of waste, downtime, and customer-centric value loss.

A. Ontology

Users, experts, and programmers may all speak the same language thanks to ontology, which facilitates communication and collaboration across all aspects of a given domain. The study of ontology is concerned with naming characteristics and providing justifications for specific assessments of those characteristics. The ontology established in this study is immune system-based and was designed for use on factory floors, however its applicability is broad enough to include the entire production facility. The created ontology contains manufacturing process metrics directly related to the functioning of machines/work stations of the floor shop, but it does not provide any details about indirect parameters such as inventory control, worker absenteeism, raw material quality, etc. To that end, it seeks to address the control data necessary for proper workstation operation and finalization of the process.

B. Cell Agents

The bits and matrices that make up a cell agent's memory hold all the data necessary to run the factory. Real and fictitious components of the production system are represented. SCADA or MES data consists of both physical and digital components, making it an essential controller/monitoring tool. The manufacturing process serves as its habitat, and its behavior is to exchange data with APC and antigens (both real and virtual). The cell agent's job is to keep the other two agents informed of their current state so that they can detect and respond to any disruptions.

C. APC and Antigen Agents

An antigen agent will produce matrices tied to actual disruption, whereas an APC agent will supply matrices of all possible virtual disruptions. The database shared by APC and Antigen Agent contains the sets of prospective disruptions, regardless of whether they are actual or virtual. In addition, the database refreshes and stores any unplanned disruption that occurs in the production line. Cell agents are provided with status updates by both the APC as well as the antigen agent. The purpose of the antigen and APC agents, which are depicted in Figures 2 and 3, is to identify any information in genuine or fictional tags that is incorrectly matched or otherwise aberrant.

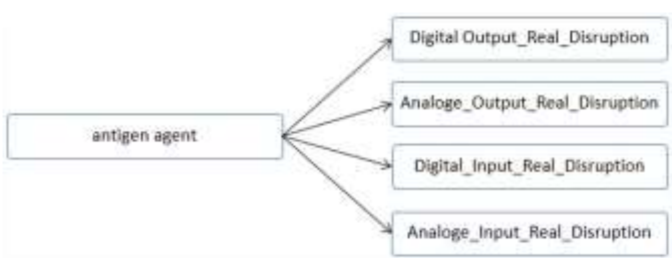

Figure 2: Antigen Agent in Protégé

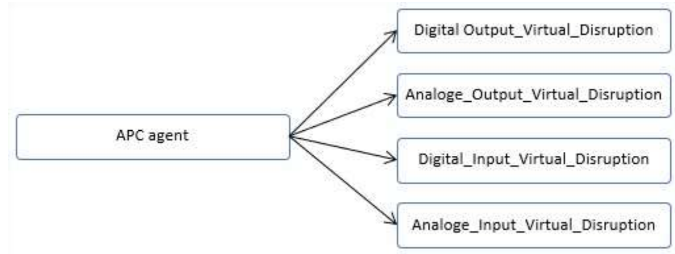

Figure 3: APC Agent Instances in Protégé

D. B and T Agents

The APC (antigen-presenting cell) agent in an immune-based disruption handling system is responsible for maintaining a database of all possible virtual disruptions, as well as any actual disruptions that occur in the manufacturing line. The APC agent also provides matrices of these disruptions to the cell agents, which are responsible for monitoring the status of the manufacturing line. The antigen agent, on the other hand, is responsible for identifying actual disruptions in the manufacturing line and providing matrices of these disruptions to the cell agents. Together, the APC and antigen agents help to identify any mismatched or abnormal information in both real and virtual tags, allowing for a more effective response to disruptions in the manufacturing process.

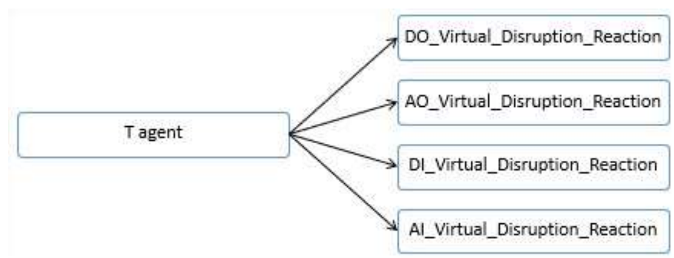

Figure 4: T Agent Instances in Protégé

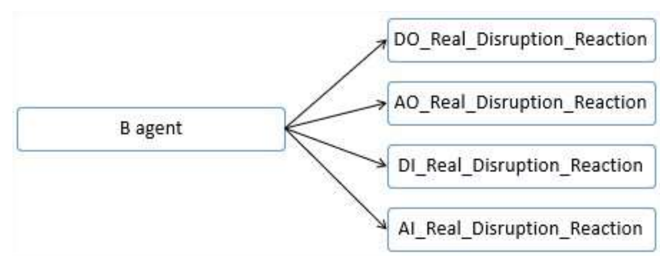

Figure 5: B Agent Instances in Protégé

E. Antibody Agent

It is the job of the antibody agent to cause the proper response to be made in response to a given disturbance. According to the proposed model, it serves as the reactant identification agent most suitable for the reaction in question. It's possible that the disturbance would be stationary, but it's also possible that it may be mobile and spread to neighboring stations. Antibody agents are used to perform a thorough data search and eliminate unnecessary reactions. The B and T agent reactions are discussed in Fig. 6. It is made clear in the diagram how the antibody agent coordinates its response with the B and T agents.

Figure 6: Anti-Body Agent Instances

F. HISFDH Modelling Formulation

The purpose of the proposed methodology that is depicted in Figure 7 is to identify and address potential faults in a manufacturing line. This is accomplished by contrasting the expected behavior of two modeled agents, the Antigen agent, and the APC agent, with the behavior that is actually exhibited by the cell agent. The method entails keeping an eye out for and evaluating any peculiar patterns in the behavior of the cell agent so that any possible problems can be pinned down to particular locations. After that, the modelled agents, B and T, are compared to previously defined fault sets so that a suitable reaction can be developed to tolerate the fault. To process reaction redundancy, using antibody agents and weights per the current state of the process have been applied. The data that was processed by the antibody agent are then compared by the fault-tolerant agent, which results in the generation of the final response. Figure 7, which depicts a flowchart, illustrates the proposed technique. This methodology consists of five stages, all of which are displayed in the flowchart.

- Detect
- Identify
- Evaluate
- Coordinate
- Validate

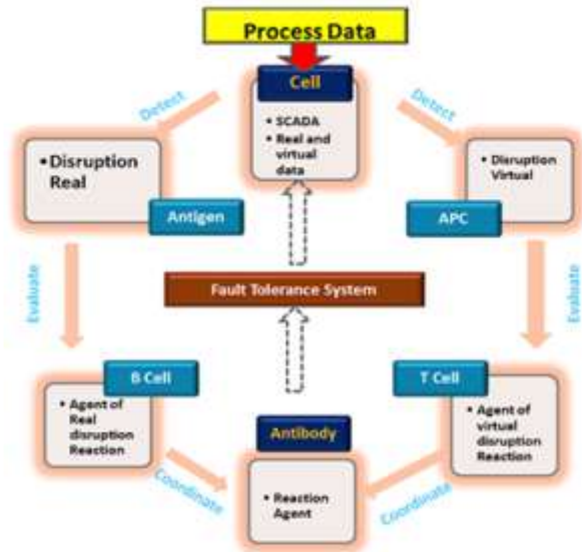

Figure 7: Data Flow in Proposed Methodology

There is a numerical value that represents the reaction (RV). The appropriate control action is determined by looking up this response value in a database. There are three possible results from the system. In the event of a critical defect or disruption, the process status could be changed to "stopped." Second, it has the ability to "resume its running condition for some time and then halt it" in the event of a defect or disruption of medium severity, protecting against sudden shutdowns of the process and so reducing the number of rejections. Third, "keep on executing the procedure" is represented by the RV value of 3. This again helps to keep things running smoothly without interruptions, which is great for both productivity and customer satisfaction. The proposed method's symbols and their meanings are laid forth in Table 1 below.

TABLE I: Proposed Methodology Symbol

Symbol	Details	Symbol	Details
W_s	Work Station	CA_r	Cell agent with real tags
r	Real Tag	CA_v	Cell agent with virtual tags
v	Virtual Tag	ABr	Antibody real
Fr	Real Disruption	ABv	Antibody virtual
Fv	Virtual Disruption	W_{ab}	Antibody weights
Ab	Antibody instances	ra	Real tag alarms or trip points
Ap	APC agent instances	va	Virtual tag alarm trip points
B	B agent instances	W_V	Final weight Virtual disruption
T	T agent Instances	W_R	Final weight for Real disruption

V. RESULTS AND DISCUSSION

A. Detection

The key and lock mechanism is used by the biological immune system to identify the invading pathogen. Where an antigen (a particular protein) on the pathogen's surface allows it to be identified. Activated cell-mediated innate immunity (APC) cells have unique receptors on their surface (one for each distinct antigen) to identify and bind to that pathogen. Here, the data collected from the factory floor is used to simulate the biological process in question. The term "cell agent" describes the nature of this information. The cell agent stores both real and simulated information, whereas the APC and Antigen agents store alarm matrices from the production process (real and virtual). Each tag has an associated alert value, as seen in (4) and (5). The difference between cell agent data and APC and antigen agent data reveals the disruption. Checks are made at each iteration of the control logic. Every data from the process is provided mathematically in terms of both virtual and physical tags. To provide for the potential of both sequential and parallel processing, the entire manufacturing facility is broken down into a variety of different workstations, as shown in equation (1).

Cell Agent,CA={W_s 1,W_s 2,W_s 3……..W_s n} (1)

The process tags are separated into two distinct categories by the cell agent: actual and virtual. "n" stands for the total number of workstations, and "m" indicates the number of virtual and actual tags that are contained within these workstations. "CA r" and "CA v" are the identifiers that are used for the real and virtual tags found in the workstations, respectively. The inputs and outputs (I/Os) of the workstations are represented by columns in the cell agent of the spreadsheet.

$$CA_r = \begin{bmatrix} r_{11} & r_{21} & . & r_{n1} \\ r_{12} & r_{22} & . & r_{n2} \\ . & . & . & . \\ . & . & . & . \\ r_{1m} & r_{2m} & . & rnm \end{bmatrix} (2)$$

$$CA_v = \begin{bmatrix} v_{11} & v_{21} & . & v_{1n} \\ v12 & v22 & . & v2n \\ . & . & . & . \\ . & . & . & . \\ vm & v2m & . & vnm \end{bmatrix} (3)$$

$$Antigen = \begin{bmatrix} ra11 & ra21 & . & ra1n \\ ra12 & ra22 & . & ra2n \\ . & . & . & . \\ . & . & . & . \\ ra1m & ra2m & . & ranm \end{bmatrix} (4)$$

$$APC = \begin{bmatrix} va11 & va21 & . & va1n \\ va12 & va22 & . & va2n \\ . & . & . & . \\ . & . & . & . \\ va1m & va2m & . & vanm \end{bmatrix} (5)$$

Disruptions in a production system can be detected in this way by employing an AND gate to compare condition (2) with condition (4) and condition (3) with condition (5).

CAr AND Antigen and CAv AND APC→ → Fr FrFrand Fv (6)

The two resulting matrices are called real disruption identified (Fr) and virtual disruption detected (Fv). They are processed further for defect identification and updated at each scan cycle.

B. Identification

If problems have been found, appropriate corrective or preventative actions must be taken. A practical solution is provided by the presented model. In the biological immune system, B-cells secrete antibodies that fight off invaders and repair injured cells. When T-cells fuse with B-cells, they send a signal via interleukin that causes the B-cell colony to grow.

In this strategy, synthetic B-cells (B-Agents) are deployed to detect and report "antigen agent flaws," while synthetic T-cells (T-Agents) are tasked with keeping an eye on antigen-presenting cells. Since equations (7) and (8) express identity matrices, the B and T agents can detect and pinpoint the source of a problem by associating "1s" with only the tags that are in error

$$B(Cell) = \begin{bmatrix} 1 & 1 & . & 1 \\ 1 & 1 & . & 1 \\ . & . & . & . \\ . & . & . & . \\ 1 & 1 & . & 1 \end{bmatrix} \quad (7)$$

$$T(Cell) = \begin{bmatrix} 1 & 1 & . & 1 \\ 1 & 1 & . & 1 \\ . & . & . & . \\ . & . & . & . \\ 1 & 1 & . & 1 \end{bmatrix} \quad (8)$$

Fault identification is achieved by dot product between B and Fr, T and Fv with

Fr Dot B=Fr.B=ABr(Antibody Real) (9)

Fv Dot T=Fv.T=ABv(Antibody Virtual) (10)

The two matrices generated through dot product are presented to anti body agent.

G. Coordination

The antibody's job is to attach to the infection and neutralize or kill it. After determining if a fault is simulated or real, the suggested method moves forward with fixing it. Antibody agent is shown it to determine if the problem is static or dynamic. As seen in Fig. 7, after disturbances have been discovered, they are coordinated between B and T responses and given weights. It takes in matrices that are the results of multiplying the dot product of B and T with the matrices for the real and virtual faults, as shown in equation (9), and (10).

In mathematics, an antibody agent can be written as a weighted matrix. Every manufacturing process tags come with their own related weights. The MTTR value of the process is used as a basis for creating a weight matrix.

Mean Time to Repair (MTTR) equals (total downtime) divided by 1. (number of breakdowns). The tag weight matrix, calculated using MTTR as described in (11).

$$W_{ab} = \begin{bmatrix} W11 & W21 & . & W1n \\ W12 & W22 & . & W2n \\ . & . & . & . \\ . & . & . & . \\ W1m & W2m & . & Wnm \end{bmatrix} (11)$$

To obtain the weights of only the faulty tags that occurred during the process, the dot product between the weight matrices (11) and the antibody matrices (9) and (10) is computed. This is done using equations (12) and (13). The resulting product gives the weight of each fault set in the process. This weight is then used to determine the level of fault tolerance required for the process. By comparing the weight of the fault set to a pre-defined threshold, the fault-tolerant agent can determine the appropriate response to the fault, whether it requires a simple correction or a more complex reaction. This helps to ensure that the manufacturing process remains efficient and effective, with minimal downtime or disruptions.

ABr Dot W_ab=ABr.W_ab=W_R (12)

(Anti body weight value for real tag)

ABv Dot W_ab=ABv.W_ab=W_V (13)

(Anti body weight value for virtual tag)

W_R and W_V Are the final values obtained through anti body agent that are forwarded to fault tolerant agent.

H. Finalization

In response to the completion of the antibody agent reaction, the Fault Tolerant (FT) agent is activated. First, the reaction is presented to FT, which takes in data from the antibody and continuously monitors the reaction's aftereffects. The threshold for assigning weights is set, and from then on, each determined response receives some weight to be used against any reaction. If the value is less than the threshold, the answer can be started; otherwise, the next available response is chosen.

Each workstation's process status, whether active or inactive, is determined by a fault-tolerant agent. Process status vector (14) is an example of this type of vector.

Ps (Process Status) = {p1,p2,….pn} (14)

The weights of malfunctioning workstations, Pw, are introduced by the fault-tolerant agent in a separate step (equation) (15).

Pw={W1,W2,W3……..Wn} (15)

Dot multiplication of ABr by Pw is used to give affected workstations their proper weights, as demonstrated in (10).

Ft(Fault tolerant)=ABr.Pw (16)

Ft(Fault tolerant)=ABv.Pw (17)

Since

ABr= {(1,&if faulty workstation @0,&if workstation has no fault)┤ And

ABv= {(1,&if faulty workstation @0,&if workstation has no fault)┤

The last step in (16) and (17) assign weights only to the faulty workstations.

The MTRR value, which is the sum of the faulty tag weight and the workstation weight, is critically important for determining the response value (finalize reaction) of fault. The process threshold value for a workstation's defects is presented in the equation together with all the factors that contribute to it (18).

RV(i)=(αFt(i)-βW_(R,V) (i)) (18)

The equation (13) is used to calculate the final response value (RV(i)) based on the values of ABr, ABv, and Ft. The tuning parameters β and α are used to adjust the weights of the faulty tags and the response value. The faulty station's weights (Ft) are calculated using equation (11) and represent the weights of only the faulty tags that occurred during the process. The final response value (RV(i)) is a key factor in determining whether the process will continue running or be stopped. The response value is calculated based on alarm trip points, which are pre-set values that indicate when an alarm should be raised to alert operators of potential issues. If the response value is below the alarm trip point, the process will continue running. If the response value exceeds the alarm trip point, an alarm will be raised, and the process will be stopped to prevent further damage or issues.

Depending on the value of the parameter, the significance of process weightage and process status will change. These parameters are adjusted for various cases. The final tolerance value will be influenced more by the aforementioned process parameters as increases. In a similar vein, the value of modifies the significance of tag weighting; a higher value increases the influence of tag on response value. At long last, the full data flow cycle has been completed. What you see in Fig.8 is how the response is formed.

Hence, this method yields three distinct classes of response values (RV). The high critical level that triggers "stopping the process" describes the maximum value for a specific defect. To explain, consider (17), which states that a faulty station has a greater impact on the process than a faulty tag, but that the latter may be safely ignored. This high critical number serves as a trip point, beyond which the system must shut down. When the RV value is at its lowest, it indicates that a malfunctioning workstation has so little impact on the whole that it may be safely disregarded. The tag is heavily weighted, but the workstation is not a crucial part of the process as a whole. The resulting action is to "continue processing". Certain situations may call for a response of "continue the state of running for some time, then halt it," which occurs when the RV value is in the medium range and is thus classified as medium level critical. In this example, the malfunctioning station and faulty tag have a moderate effect on the entire operation

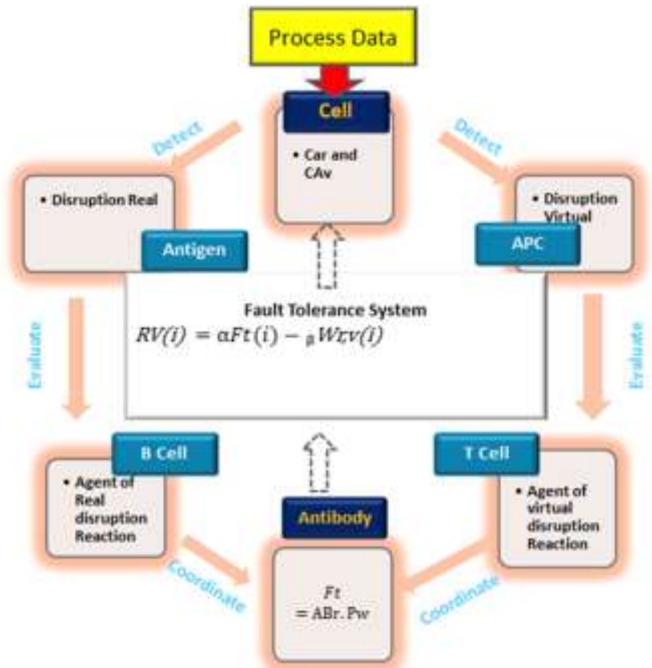

Figure 8: Flow Chart for HISFDH

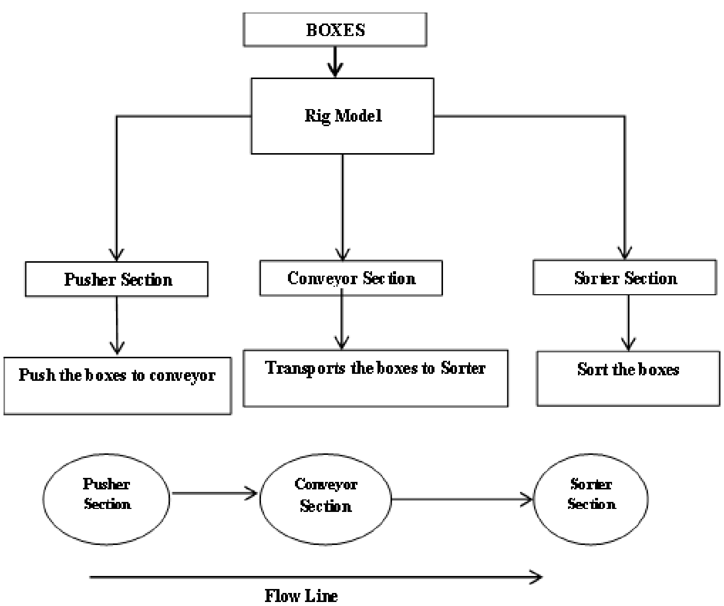

Figure 9: Flow Chart for entire process

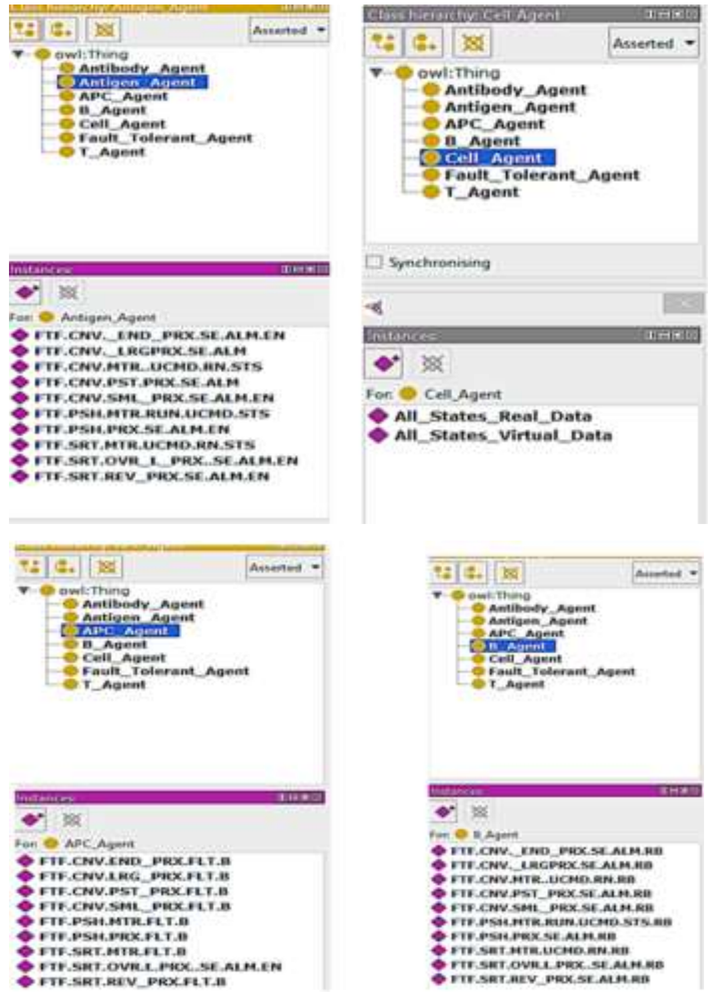

Figure 10: Agent instances for test bed

I. *Analyzing established control logic*

Methods like the HISFDH and the Fault and Behavioral Anomaly Detection Tool for Manufacturing Process (proposed by Gosh et al. After 24 hours of operation, the process has experienced various disruptions due to various causes, which have had an effect on downtime, consumer-centric value, and process efficiency. In Table 1 and Fig. 12, we can see the breakdown of the disruptions and the downtime in minutes. The downtime analysis of the process after 24 hours of nonstop operation is presented in Fig. 9, which provides a detailed breakdown of the downtime for each workstation. According to the analysis, the estimated downtime for the entire operation is 139 minutes (2.12 hours). According to the findings of the investigation, the Conveyer End Position Proximity Signal Error Alarm Enable (C.P.P.S.E.A.E), which resulted in a downtime of 53 minutes, is the parameter that is most likely to be held accountable for the downtime that occurred at workstation 2 (the conveyor section). The Conveyer Position Proximity Sensor Detect Status (C.P.P.S.D.S) at workstation 1 is the second most likely parameter that is responsible for downtime. This particular parameter generated a downtime of thirteen minutes. The Conveyor Actuator Run Command and the Conveyor Actuator Run Status are two additional characteristics that each produced a downtime of about 9 minutes and contributed to the disturbances that occurred throughout the process. 11 minutes of unscheduled downtime were incurred as a result of the Conveyor Small Proximity Sensor Signal Error Alarm Enable (C.S.P.S.S.E.A.E) error.

Table 2: Downtime analysis of tags using FBMTP

Tags/IOs of three Sections	Symbol	Down Time(minutes)
Conveyer End Position Proximity signal error alarm enable	C.P.P.S.E.A.E	20
Conveyer Position Proximity sensor detect status	C.P.P.S.D.S	13
Conveyer Small Proximity sensor signal error alarm enable	C.S.P.S.S.E.A.E	11
Conveyer Actuator Run Command	C.A.R.C	11
Conveyer Actuator Run status	C.A.R.S	9
Conveyer Actuator uncommanded Run Status	C.A.U.R.S	7
Conveyer End Position Proximity sensor detect command	C.P.P.S.D.C	6
Conveyer End Position Proximity sensor detect status	C.P.P.S.D.S	5
Conveyer End Position Proximity sensor Fault Bit	C.P.P.S.F.B	5
Conveyer End Position Proximity sensor Fault Reaction Bit	C.P.P.S.F.R.B	5
Conveyer End Position Proximity sensor Not detect status	C.P.P.S.N.D.S	4
Conveyer Actuator Fault Bit	C.A.F.B	4
Conveyer Large Position Proximity sensor detect command	C.L.P.P.S.D.C	4
Conveyer Large Position Proximity sensor detect status	C.L.P.P.S.D.S	4
Conveyer Large Position Proximity sensor Fault Bit	C.L.P.P.S.F.B	4
Conveyer Large Position Proximity sensor Not detect status	C.L.P.P.S.N.D.S	3
Conveyer Large Proximity sensor signal error alarm enable	C.L.P.S.S.E.A.E	3
Conveyer Position Proximity sensor detect command	C.P.P.S.D.C	3
Conveyer Actuator Fault Reaction Bit	C.A.F.R.B	3
Conveyer Position Proximity sensor Fault Bit	C.P.P.S.F.B	3
Conveyer Position Proximity sensor Fault Reaction Bit	C.P.P.S.F.R.B	3
Conveyer Position Proximity sensor Not detect status	C.P.P.S.N.D.S	2
Conveyer Actuator Not Run status	C.A.R.N.S	2
Conveyer Small Proximity sensor signal error alarm enable	C.S.P.S.S.E.A.E	2
Conveyer Small Position Proximity sensor detect command	C.S.P.P.S.D.C	2
Conveyer Small Position Proximity sensor detect status	C.S.P.P.S.D.S	1
Conveyer Small Position Proximity sensor Fault Bit	C.S.P.P.S.F.B	0
Conveyer Small Position Proximity sensor Fault Reaction Bit	C.S.P.P.S.F.R.B	0
Conveyer Small Position Proximity sensor Not detect status	C.S.P.P.S.N.D.S	0

In conclusion, the findings of the investigation indicate that the Control Air Regulator Station (CARS) and the Control Air Unit Run Station (CAURS) had the least amount of downtime overall, with 9 and 7 minutes, respectively. In general, the comprehensive downtime analysis is beneficial in that it assists in the identification of the essential parameters that need to be addressed in order to improve the effectiveness of the process and reduce downtime.

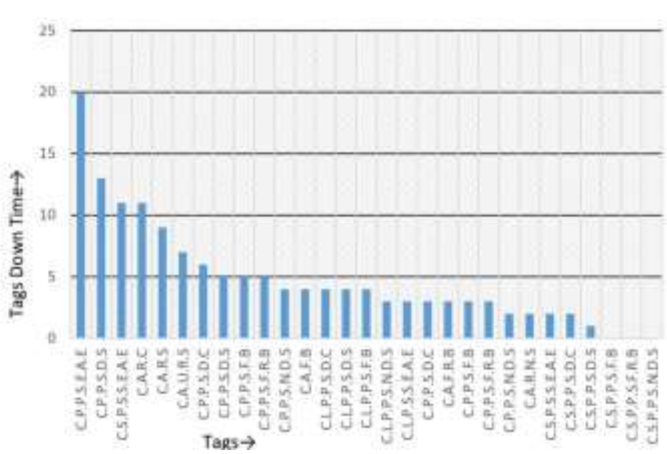

Figure 11: Process downtime in 24 hours

Moreover, the data contains numerous disruption types that can be found in nearly every tag shown in Fig.10. It analyses the top five significant tags that account for over 50.0% of the total downtime caused by disruption. There are many causes of disruption in this case, but the top five are accounted for in the table above. As they have such a negative effect on the proposed model's consumer-centric value and efficiency, they are the model's primary concern.

J. Implementation of the proposed HISFDH model

It is run under the same settings as before after the suggested HISFDH model, S7-1200 PLC, and WinCC Flexible have been incorporated into the test bench. To evaluate the efficacy of the methodology, scripting capabilities were included in the SCADA software that was utilized. When the framework was proposed, the primary emphasis was placed on stimulating parameters such as C.P.P.S.D.S. and C.S.P.S.E.E.E. After a period of rest for twenty-four hours, the test bed was utilized once more. The cumulative amount of downtime was cut by 50 minutes, bringing the total to 82 minutes (see Table 3 and Fig.12). It is important to note that the anticipated results were much higher for the factors that were the focus of the study. C.P.P.S.D.S., for instance, was responsible for a 10-minute decrease, going from 13 minutes down to 3, whereas C.P.P.S.E.A.E. had the most significant downturn of 16 minutes, going from 20 minutes down to 4, and C.S.P.S.E.A.E., C.A.R.C., and C.A.R.S. were each noticed with a 5-minute decrease, going from 11 minutes On the other hand, the factors that had a less significant impact on downtime in the past showed a tendency in the opposite direction. For instance, the amount of time that the C.P.P.S.N.D.S. downtime interruption occurs has increased by 4 minutes, the amount of time that it occurs at C.P.P.S.D.C. has grown by 5 minutes, and the amount of time that it occurs at C.L.P.P.S.F.B. has increased by 10 minutes. Table 3 and the accompanying figure provide irrefutable evidence that the proposed architecture drastically cuts downtime, dramatically boosts process efficiency, and places a priority on adding value for consumers.

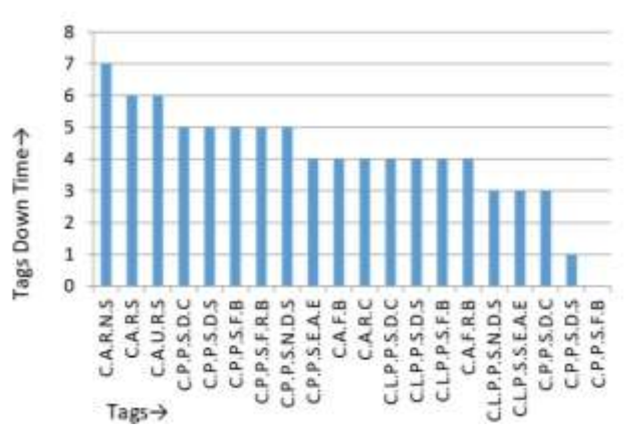

Figure 12: Minimal HISFDH-Related Process Downtime (in Minutes)

Table 3: Process downtime on implementation of HISFDH

Description	Symbol	Down Time(minutes)
Conveyer Actuator Not Run status	C.A.R.N.S	7
Conveyer Actuator Run status	C.A.R.S	4
Conveyer Actuator uncommented Run Status	C.A.U.R.S	6
Conveyer End Position Proximity sensor detect command	C.P.P.S.D.C	5
Conveyer End Position Proximity sensor detect status	C.P.P.S.D.S	5
Conveyer End Position Proximity sensor Fault Bit	C.P.P.S.F.B	5
Conveyer End Position Proximity sensor Fault Reaction Bit	C.P.P.S.F.R.B	5
Conveyer End Position Proximity sensor Not detect status	C.P.P.S.N.D.S	5
Conveyer End Position Proximity signal error alarm enable	C.P.P.S.E.A.E	4
Conveyer Actuator Fault Bit	C.A.F.B	4
Conveyer Actuator Run Command	C.A.R.C	4
Conveyer Large Position Proximity sensor detect command	C.L.P.P.S.D.C	4
Conveyer Large Position Proximity sensor detect status	C.L.P.P.S.D.S	4
Conveyer Large Position Proximity sensor Fault Bit	C.L.P.P.S.F.B	4
Conveyer Actuator Fault Reaction Bit	C.A.F.R.B	3
Conveyer Large Position Proximity sensor Not detect status	C.L.P.P.S.N.D.S	3
Conveyer Large Proximity sensor signal error alarm enable	C.L.P.S.S.E.A.E	6
Conveyer Position Proximity sensor detect command	C.P.P.S.D.C	3
Conveyer Position Proximity sensor detect status	C.P.P.S.D.S	1
Conveyer Position Proximity sensor Fault Bit	C.P.P.S.F.B	0

VI. CONCLUSION

In this research, we used an artificial immune system to propose and prove a preferable solution to the problem of production disruption. After 24 hours of data collection, ontologies were first built, then deployed (in a lab setting). The results were satisfactory in that just 24 percent of all disruptions occurred in 24 hours, and only 36 minutes of those were major (targeted) disruptions. After HISFDH was introduced, the efficiency of the process increased by 5%. In addition, the process efficiency was improved by a whopping 31% due to the decreased amount of downtime experienced in the five focus areas. There was a 5.1% increase in output as a result of this change. As a result, the experimental apparatus had successfully categorized an additional 216 containers. The potential for the proposed model to improve the plant's efficiency is promising.

Further applications of the framework include the ability to clearly illustrate a variety of disruption management difficulties across a wide range of production factors, including stockouts, shortages of raw materials, and labor shortages. The model can also be improved to take into account the alignment of workstations in the flow path.

VII. REFERENCES

[1] A. Hofmeyr, Steven, and S Forrest. 1999. "Architecture for an Artificial Immune System." Evolutionary Computation 7: 45–68.

[2] Alenljung, Tord, Markus Sköldstam, Bengt Lennartson, and Knut Åkesson. 2007. "PLC-Based Implementation of Process Observation and Fault Detection for Discrete Event Systems." Proceedings of the 3rd IEEE International Conference on Automation Science and Engineering, IEEE CASE 2007, 207–12. https://doi.org/10.1109/COASE.2007.4341752.

[3] Anchor, KP, JB Zydallis, and GH Gunsch. 2002. "Extending the Computer Defense Immune System: Network Intrusion Detection with a Multiobjective Evolutionary Programming Approach." In ICARIS 2002: 1st International Conference on Artificial Immune Systems Conference Proceedings, 12–21. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.9.6943&rep=rep1&type=pdf.

[4] Aytug, Haldun, Mark A. Lawley, Kenneth McKay, Shantha Mohan, and Reha Uzsoy. 2005. "Executing Production Schedules in the Face of Uncertainties: A Review and Some Future Directions." European Journal of Operational Research 161 (1): 86–110. https://doi.org/10.1016/j.ejor.2003.08.027.

[5] Azzini, Antonia, Ernesto Damiani, Gabriele Gianini, and Stefania Marrara. 2009. "An Ontology for Artificial Immune Systems." 2009 3rd IEEE International Conference on Digital Ecosystems and Technologies, DEST '09, 324–28. https://doi.org/10.1109/DEST.2009.5276747.

[6] Bao, Jian, Huifeng Wu, and Yimajian Yan. 2014. "A Fault Diagnosis System-PLC Design for System Reliability Improvement." International Journal of Advanced Manufacturing Technology 75 (1–4): 523–34. https://doi.org/10.1007/s00170-014-6166-z.

[7] Battini, Daria, Martina Calzavara, Alena Otto, and Fabio Sgarbossa. 2017. "Preventing Ergonomic Risks with Integrated Planning on Assembly Line Balancing and Parts Feeding." International Journal of Production Research 7543: 1–21. https://doi.org/10.1080/00207543.2017.1363427.

[8] Bayar, Nawel, Saber Darmoul, Sonia Hajri-Gabouj, and Henri Pierreval. 2015. "Fault Detection, Diagnosis and Recovery Using Artificial Immune Systems: A Review." Engineering Applications of Artificial Intelligence 46 (August): 43–57. https://doi.org/10.1016/j.engappai.2015.08.006.

[9] Bruccoleri, Manfredi, Zbigniew J. Pasek, and Yoram Koren. 2006. "Operation Management in Reconfigurable Manufacturing Systems: Reconfiguration for Error Handling." International Journal of Production Economics 100 (1): 87–100. https://doi.org/10.1016/j.ijpe.2004.10.009.

[10] Castro, Leandro N. De, and Fernando J. Von Zuben. 2002. "Learning and Optimization Using the Clonal Selection Principle." IEEE Transactions on Evolutionary Computation 6 (3): 239–51. https://doi.org/10.1109/TEVC.2002.1011539.

[11] Castro, Leandro Nunes de, and Fernando J. Von Zuben. 2000. "The Clonal Selection Algorithm with Engineering Applications." Proceedings of GECCO, 36–39.

[12] Cauvin, A. 2005. "Analyse, Modélisation et Amélioration de La Réactivité Des Systèmes de Décision Dans Les Organisations Industrielles." Habilitation à Diriger Des Recherches.

[13] Darmoul, Saber, Henri Pierreval, and Sonia Hajri-Gabouj. 2011. "Using Ontologies to Capture and Structure Knowledge about Disruptions in Manufacturing Systems: An Immune Driven Approach." IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, no. September 2011. https://doi.org/10.1109/ETFA.2011.6059033.

[14] Darmoul, Saber, Henri Pierreval, and Sonia Hajri-Gabouj. 2013. "Handling Disruptions in Manufacturing Systems: An Immune Perspective." Engineering Applications of Artificial Intelligence 26 (1): 110–21. https://doi.org/10.1016/j.engappai.2012.09.021.

[15] Darmoul, Saber, Henri Pierreval, and Sonia Hajri–Gabouj. 2011. "An Immune Inspired Multi Agent System to Handle Disruptions in Manufacturing Production Systems." International Conference on Industrial Engineering and Systems Management, IESM'2011, no. May.

[16] Dasgupta, D, and S Forres. 1999. "Artificial Immune Systems in Industrial Applications." Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials, 257–67.

[17] Dasgupta, D, and S Forrest. 1995. "Tool Breakage Detection in Milling Operations Using a Negative-Selection Algorithm."

[18] Deaton, R., M. Garzon, J.A. Rose, R.C. Murphy, S.E., Jr. Stevens, and D.R. Francheschetti. 1997. "A DNA Based Artificial Immune System for Self-Nonself\ndiscrimination." 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation 1: 862–66. https://doi.org/10.1109/ICSMC.1997.626210.

[19] Forrest, Stephanie, Alan S. Perelson, Lawrence Allen, and Ra jesh Cherukuri. 1994. "Self-Nonself Discrimination in a ComputerNo Title." Research in Security and Privacy, 1994. Proceedings, 202–12.

[20] Fukuda, T, K Mori, and M Tsukiama. 1999. "Parallel Search for Multi-Modal Function Optimization with Diversity and Learning of Immune Algorithm." Artificial Immune Systems and Their Applications, 210–20. https://doi.org/10.1007/978-3-642-59901-9_11.

[21] Ghosh, Arup, Jooyeoun Lee, and Gi-nam Wang. 2016. "PLAT : An Automated Fault and Behavioural Anomaly Detection Tool for PLC Controlled Manufacturing Systems" 2016: 1–27.

[22] Ghosh, Arup, Shiming Qin, Jooyeoun Lee, and Gi Nam Wang. 2017. "FBMTP: An Automated Fault and Behavioral Anomaly Detection and Isolation Tool for PLC-Controlled Manufacturing Systems." IEEE Transactions on Systems, Man, and Cybernetics: Systems 47 (12): 3397–3417. https://doi.org/10.1109/TSMC.2016.2633392.

[23] Greensmith, Julie, and Steve Cayzer. 2003. "An Artificial Immune System Approach to Semantic Document Classification." Artificial Immune Systems, 136–37. https://doi.org/10.1145/1066677.1066889.

[24] Grubic, Tonci, and Ip Shing Fan. 2010. "Supply Chain Ontology: Review, Analysis and Synthesis." Computers in Industry 61 (8): 776–86. https://doi.org/10.1016/j.compind.2010.05.006.

[25] Hajej, Zied, Nidhal Rezg, and Ali Gharbi. 2018. "Quality Issue in Forecasting Problem of Production and Maintenance Policy for Production Unit." International Journal of Production Research 56 (18): 6147–63. https://doi.org/10.1080/00207543.2018.1478150.

[26] Hart, E., P. Ross, and J. Nelson. 1998. "Producing Robust Schedules via an Artificial Immune System." 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), 464–69. https://doi.org/10.1109/ICEC.1998.699852.

[27] Hu, W., A. G. Starr, and A. Y.T. Leung. 1999. "Two Diagnostic Models for PLC Controlled Flexible Manufacturing Systems." International Journal of Machine Tools and Manufacture 39 (12): 1979–91. https://doi.org/10.1016/S0890-6955(99)00022-X.

[28] Hu, W., A. G. Starr, and A. Y.T. Leung. 2003. "Operational Fault Diagnosis of Manufacturing Systems." Journal of Materials Processing Technology 133 (1–2): 108–17. https://doi.org/10.1016/S0924-0136(02)00252-2.

[29] Huang, Sunan, Kok Kiong Tan, and Mingbo Xiao. 2015. "Automated Fault Diagnosis and Accommodation Control for Mechanical Systems." IEEE/ASME Transactions on Mechatronics 20 (1): 155–65. https://doi.org/10.1109/TMECH.2014.2322652.

[30] Ishida, Yoshiteru. 1993. "An Immune Network Model and Its Applications to Process Diagnosis." Systems and Computers in Japan 24 (6): 38–46. https://doi.org/10.1002/scj.4690240604.

[31] Ishiguro, a., Y. Watanabe, and Y. Uchikawa. 1994. "Fault Diagnosis of Plant Systems Using Immune Networks." Proceedings of 1994 IEEE International Conference on MFI '94. Multisensor Fusion and Integration for Intelligent Systems, 34–42. https://doi.org/10.1109/MFI.1994.398475.

[32] Klein, Stéphane, Lothar Litz, and Jean-jacques Lesage. 2014. "fault detection of discrete event systems using an identification approach." Quality, no. July 2005.

[33] Knight, T., and J. Timmis. 2001. "AINE: An Immunological Approach to Data Mining." Proceedings 2001 IEEE International Conference on Data Mining, 297–304. https://doi.org/10.1109/ICDM.2001.989532.

[34] Leandro Nunes de, Castro, and Timmis Jonathan. 2002. Artificial Immune Systems: A New Computational Intelligence Approach.

[35] Liu, Tongshun, Kunpeng Zhu, and Liangcai Zeng. 2018. "Diagnosis and Prognosis of Degradation Process via Hidden Semi-Markov Model." IEEE/ASME Transactions on Mechatronics 23 (3): 1456–66. https://doi.org/10.1109/TMECH.2018.2823320.

[36] Machado, Alencar, Daniel Lichtnow, Ana Marilza Pernas, Amel Bouzeghoub, Iara Augustin, and Leandro Krug Wives. 2015. "A Framework Reactive and Proactive for Pervasive Homecare Environments" 2: 320–38. https://doi.org/10.1007/978-3-319-22348-3.

AUTHORS

Zain Abbas: Holds a Bachelor's in Electrical Engineering from the Comsats University of Science and Technology, Islamabad. Currently pursuing a Master's Program in Mechatronics Engineering from UET Peshawar.

Sher Shah: Holds a Bachelor's in Mechatronics Engineering from the University of Engineering and Technology, Peshawar. Currently pursuing a Master's Program in Industrial Management and Innovation from Uppsala University, Sweden.

Zubair Ahmad Khan: Holds a Ph.D. in Mechatronics from the University of Engineering & Technology, Peshawar, Pakistan. Completed an MS in Mechanical Engineering from the University of Engineering & Technology, Peshawar, and a BE in Mechatronics Engineering from the National University of Science and Technology, Pakistan. Currently serving as a Lecturer in the Department of Mechatronics Engineering at the University of Engineering & Technology, Peshawar.

Hamza Ahmad Khan: Holds a Master's in Mechatronics Engineering from the University of Engineering and Technology, Peshawar.

Correspondence Author – **Zain Abbas**