IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 6 June 2023, accepted 17 June 2023, date of publication 26 June 2023, date of current version 29 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3289536

RESEARCH ARTICLE

3D-AmplifAI: An Ensemble Machine Learning Approach to Digital Twin Fault Monitoring for Additive Manufacturing in Smart Factories

GABRIEL AVELINO R. SAMPEDRO[1,2], (Member, IEEE), MADE ADI PARAMARTHA PUTRA[3], AND MIDETH ABISADO[4], (Member, IEEE)

[1]Faculty of Information and Communication Studies, University of the Philippines Open University, Los Baños 4301, Philippines
[2]Center for Computational Imaging and Visual Innovations, De La Salle University, Malate, Manila 1004, Philippines
[3]Informatic Engineering, STMIK Primakara, Denpasar, Bali 80226, Indonesia
[4]College of Computing and Information Technologies, National University, Manila 1008, Philippines

Corresponding author: Gabriel Avelino R. Sampedro (garsampedro@ieee.org)

ABSTRACT In the digital age, the digital twin eliminates physical barriers and risks, facilitating seamless activities in both real and virtual worlds. In the context of additive manufacturing, testing 3D printers can be resource-intensive and prone to printing issues. This research introduces a digital twin-based system that employs the innovative ensemble 3D-AmplifAI algorithm for fault monitoring in 3D printers. The system continuously monitors real-time temperature values and detects faults to prevent potential damage to the printer. Through an ensemble method, the 3D-AmplifAI algorithm combines multiple machine learning models to enhance fault detection in 3D printers. The digital twin environment, developed using Unity, serves as the bridge connecting the physical printer to the virtual world. Comparative evaluations against state-of-the-art algorithms, including Ridge Regression, XGBoost, InceptionTime, Time Series Transformer (TST), Rocket Ridge, Logistic Regression, Rocket XGBoost, ResNet, and Rocket Ridge Regression, demonstrate the superior performance of the 3D-AmplifAI algorithm in terms of accuracy, precision, recall, and F1-score.

INDEX TERMS Additive manufacturing, digital twin, ensemble algorithm, fault monitoring.

I. INTRODUCTION

Additive manufacturing, also known as 3D printing, has transformed the process of developing tangible components and prototypes. This technology enables the creation of three-dimensional objects by depositing successive layers of material based on digital models. Its impact spans across various industries, from healthcare to aerospace, revolutionizing the way engineers design, prototype, and manufacture products [1], [2], [3]. As a result, 3D printing is increasingly adopted in new fields, with experts predicting its continuous influence on the future of manufacturing, engineering, and design. The global shipment of 3D printers reached 2.2 million units in 2021, and it is projected to grow to 21.5 million units by 2030 [4]. However, the rise of additive manufacturing technology has also brought about challenges, particularly in terms of surface quality.

As additive manufacturing technology advances, it becomes evident that the process presents various challenges, with error handling being one of the most significant. Errors can arise from factors such as incorrect machine settings, flawed designs, and material defects [5], [6]. Identifying and rectifying these errors during the manufacturing process is crucial to avoid the production of non-functional or unsafe parts. Consequently, there is an increasing need for effective error-handling strategies capable of real-time error detection and correction, minimizing the risk of defective parts and ensuring the overall quality of the final product [7], [8]. In this context, researchers and industry professionals are actively exploring new approaches to error handling that can address the unique challenges of additive manufacturing and unlock its full potential.

The associate editor coordinating the review of this manuscript and approving it for publication was Chi-Tsun Cheng.

Currently, most research on monitoring additive manufacturing devices relies on image and sensor-based monitoring [9]. The objective of monitoring is to identify and detect irregularities that may lead to printing failures. Printing failure refers to the rejection of output by the additive manufacturing device due to surface imperfections. The irregularities being detected involve obscure parameters related to the operational conditions of the printer, such as overheating printer heads, excessive vibration, excessive noise, and underheated materials [10], [11]. Prior studies have implemented machine learning-based solutions to understand the measured values for determining the conditions of various printer parts.

However, there is still a significant research gap to be addressed by simulating test values in a controlled setting, as conducting experiments on actual 3D printers can be costly and time-consuming. Limited studies have explored the use of virtual 3D printers to enhance the simulation process. Therefore, the development of a digital twin-enabled monitoring device becomes crucial to address this issue.

Digital twin technology has emerged as a powerful tool for simulation and analysis in various industries, ranging from aerospace and automotive to healthcare and energy [12]. Essentially, a digital twin is a virtual replica of a physical object or system that enables real-time monitoring, analysis, and optimization of its performance [13]. In the context of additive manufacturing, digital twin technology holds the potential to revolutionize error handling by creating virtual replicas of the printing process that can be continuously monitored for faults and defects [14]. By comparing the digital twin with the physical process in real-time, manufacturers can identify and rectify errors before they result in defective parts, thereby reducing waste and improving efficiency. Consequently, there is a growing interest in applying digital twin technology to fault monitoring in additive manufacturing, prompting researchers and industry professionals to explore new approaches to unlock the full potential of this innovative technology [15].

The general objective of this research is to develop a digital twin-enabled 3D printer monitor capable of simulating and detecting possible errors in a 3D printer. The specific objectives include: (i) developing a digital twin of a 3D printer for monitoring and (ii) applying machine learning-based algorithms to the measurements collected from sensors attached to a 3D printer and test values set on the digital twin to detect potential faults. By implementing digital twin technology, it becomes possible to experiment with 3D printers through simulation while maintaining a connection with the physical counterpart. The major contributions of this research are detailed as follows:

- Designing a digital twin environment for smart additive manufacturing using an FDM 3D printer that accurately simulates the physical component and performs print actions.
- Proposing the ensemble 3D-AmplifAI model, which combines multiple machine learning models to achieve enhanced accuracy in detecting early fault conditions in the 3D printer.
- Conducting an extensive performance evaluation of various machine learning models with the proposed 3D-AmplifAI system, demonstrating its robustness through the use of numerous performance metrics.
- Integrating the machine learning model with the digital twin environment, enables the proposed system to mitigate faults in both the physical and virtual domains effectively.

The proposed system will provide 3D printer operators with a more graphical view of the monitoring process, enabling quick identification of error sources compared to other monitoring systems. This improved visualization will facilitate a more in-depth analysis of the root causes of issues. Furthermore, this study will lay the groundwork for further research in fault monitoring for 3D printers.

The study is organized into seven distinct sections. The first section introduces the topic and provides an overview of the subsequent chapters. Section II explores related works and reviews the literature on fault monitoring systems, machine learning algorithms, and virtual environments. Section III covers the various machine learning algorithms employed in this work and the development details of the digital twin-based fault monitoring system, which includes the design, implementation, and features of a virtual printer environment. The digital twin-based technique for prediction and simulation is detailed in Section IV. Section V outlines the experimental setup, dataset collection, and preprocessing techniques employed. Section VI presents the performance evaluation of the entire system, discussing its accuracy, efficiency, and effectiveness. Finally, the last section concludes the manuscript, summarizing the findings, and discussing future research directions in the field.

II. LITERATURE REVIEW

This section explains the previous works of researchers that proposed different methodologies to address the issue of fault monitoring in additive manufacturing, specifically for smart factories. Artificial intelligence is one of the emerging technologies in focus today, and researchers explore the application of various machine learning and deep learning approaches to monitor for faults.

A. FAULT MONITORING FOR INDUSTRY 4.0

Fault monitoring plays a crucial role in Industry 4.0 and smart factories, as highlighted by various studies. Kalsoom et al. emphasize the importance of advanced low-cost sensor technologies in data collection for effective performance by manufacturing companies and supply chains [16]. Different sensor technologies used in smart factories underscore the need for data-driven decision-making in a factory's operations, optimizing efficiency by knowing the overall conditions of the various machines.

Augmented Reality (AR) has been proposed as a concept to present information to human operators in an intuitive way in several studies. Tzimas et al. proposed AR applications for industrial guidance and training, demonstrating the technical aspects of AR application development, including interfacing distance sensors, scenario structuring, and techniques ensuring expansibility, flexibility, and ease of authoring such applications [17]. Meanwhile, Longo et al. proposed a human-centric and knowledge-driven approach to Industry 4.0 initiatives, emphasizing the importance of ubiquitous knowledge about the manufacturing system that is intuitively accessed and used by manufacturing employees. Their service-oriented digital twin prototype leverages a flexible ontology-oriented knowledge structure and AR combined with a vocal interaction system for intuitive knowledge retrieval and fruition. The study shows that a human-centric and knowledge-driven approach can drive the performance of Industry 4.0 initiatives and lead a smart factory toward its full potential [18].

These studies demonstrate that fault monitoring is critical in Industry 4.0 and smart factories. Integrating Industrial Internet of Things (IoT) concepts with additive manufacturing techniques benefit industries and material manufacturers [19]. Data-driven analysis and constant monitoring through technology are required for Industry 4.0 technologies, particularly for additive manufacturing. Advanced low-cost sensor technologies, AR applications, human-centric and knowledge-driven approaches, and IoT additive manufacturing integrated techniques can improve performance, reduce waste, and fulfill customer specifications.

Various research has been conducted to enhance the efficiency of smart manufacturing through the utilization of machine learning. For instance, Agron et al. investigated the nozzle of a fused deposition modeling printer and employed a temporal neural network with a two-stage sliding window strategy (TCN-TS-SW) to predict future thermal values of the nozzle tip [20]. Zhang et al. utilized a deep hybrid state network with feature reinforcement, leveraging data collected by an attitude sensor attached to a printer, to diagnose faults [21]. Belikovetsky et al. used digital audio signatures to analyze the sound produced by a 3D printer's stepper motors and identify irregularities that could lead to faults [22]. Liu et al. performed image analysis-based diagnosis on the captured image of the surface of the output to analyze the output of a 3D printer [23]. Similarly, a study by [24] explored efficient fault detection based on an image dataset using a multi-block Convolutional Neural Network (CNN)-based model, outperforming various pre-trained networks. Furthermore, Deepraj et al. proposed XAI-3DP, which presents a data-driven approach for fault diagnosis in 3D printers. They collected data for three scenarios, including healthy conditions, bed failure, and arm failure, and implemented an ensemble learning model of Random Forest and XGBoost [25].

B. DIGITAL TWIN-BASED FAULT MONITORING

Various researchers have explored the possibility of implementing a digital twin on additive manufacturing technology, but only on a conceptual level. Knapp et al. highlights the need to apply digital twin technology to additive manufacturing to minimize waste and improve the efficiency of the process [26]. The accurate prediction of the physical properties of the different parts of a 3D printer using digital twin technology is a critical benefit that can help to minimize waste and optimize the process.

Debroy et al. discuss the technicalities surrounding the application of digital twin technology in additive manufacturing [27]. They stress the importance of considering the application of laws in physics, such as heat transfer equations, mechanics of materials, and solidification modeling, when implementing digital twin technology. In addition, they emphasize the need to integrate data modeling algorithms further to understand a printer's behavior and future behavior. This allows for a more accurate prediction of the parts' physical properties, which can help optimize the additive manufacturing process and minimize waste.

Stavropoulos et al. and Kabaldin et al. explore the development of a digital twin for controlling a 3D printer [28], [29]. They develop a system model that integrates physics and systematics-based modeling, uncertainty quantification, and capability tracking to create a digital twin-based controller for 3D printers. Using such technology optimizes the additive manufacturing process, as a data-driven approach allows errors to be minimized. Furthermore, digital twin technology allows for a rapid response to issues, and issues in additive manufacturing may be easily mitigated.

Overall, the application of digital twin technology in additive manufacturing has the potential to improve the efficiency of the process and minimize waste significantly. By accurately predicting the physical properties of the parts being produced, errors can be minimized, and the process can be optimized. Developing a digital twin-based controller for 3D printers is a promising area of research, and we will likely see further results in this field in the coming years. To the best of our knowledge, no research has investigated the implementation of a fault detection system using an ensemble algorithm in a digital-twin system. Therefore, this study aims to provide insight into such a system's implementation to improve the efficiency of 3D printers in additive manufacturing.

III. PROPOSED SYSTEM

The problem that this study aims to address is the production waste in the process of additive manufacturing. Additive manufacturing is a rapidly growing field with immense potential, but it is challenging. One of the major challenges faced in the additive manufacturing process is the need for a monitoring system that can diagnose issues on a 3D printer and predict its behavior in a machine learning-based approach.

Traditional methods of identifying and addressing issues during manufacturing involve extensive trial-and-error testing. This approach can be time-consuming and expensive and may not always yield the desired results. Additionally, due to the systems' complexity, it can take time to identify the root cause of issues that arise during manufacturing. As a result, there is a need for a more effective and efficient approach to addressing issues in additive manufacturing.

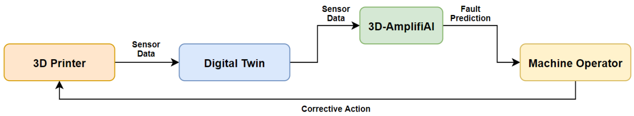

Figure 1. **The conceptual framework of this research is based on the development of a fault monitoring system for additive manufacturing using a digital twin. The framework includes the integration of hardware, software, and machine learning algorithms to create a virtual replica of the manufacturing process that enables real-time fault monitoring, diagnosis, and corrective action by the operator.**

To gain a better understanding of the problems that lie beneath the diagnostic aspects of 3D printing, fieldwork was conducted in mid-year 2020. An additive manufacturing company was examined, and the processes used in error mitigation were analyzed. It was found that the lack of real-time monitoring and predictive capabilities in 3D printing was a significant factor contributing to production waste.

Various related works by other researchers were examined, revealing that the problem of production waste in additive manufacturing is not unique to this study. Several other researchers have highlighted the need for a more efficient and practical approach to addressing issues in additive manufacturing.

Furthermore, the adoption of a digital twin-based system shows promise in minimizing the trial-and-error process before the actual printing process is conducted to obtain the best possible configuration, thus reducing the possibility of wasted printing materials.

A. MODEL CONSIDERATIONS

This case study focuses on developing a digital twin for diagnosing the operational conditions of a 3D printer and predicting its future behavior. The difficulties of a manual process have been identified in various disciplines, particularly in engineering. Additionally, older printers tend to behave abnormally over time. The stochastic behavior of printers, particularly in terms of temperature stability, can pose a problem in the printing process.

A review of comprehensive literature on the diagnosis of a 3D printer reveals a need for a conceptual framework for finding a solution that provides an in-depth analysis of the condition of a 3D printer on a digital twin-enabled system. Currently, several proposed solutions explore the application of machine learning algorithms for fault detection using sensor-based and image-based monitoring. Despite the various solutions presented, none use digital twin technology, which is essential in enabling a smart industrial system [30].

An analysis of this conceptual framework for the process of applying digital twin technology in diagnosing the future behavior of a 3D printer will not only provide vital information on the subject matter but also shed light on a concrete solution that can help machine operators mitigate the damage caused by errors.

Figure 1 shows the conceptual framework of the research. The research aims to obtain input from the different embedded sensors of a 3D printer for monitoring possible faults of a printer. From the collected data, ensemble machine learning methods were applied to anticipate possible faults given the historical data from the sensors and a dataset of sensor values collected over a period of 1000 milliseconds.

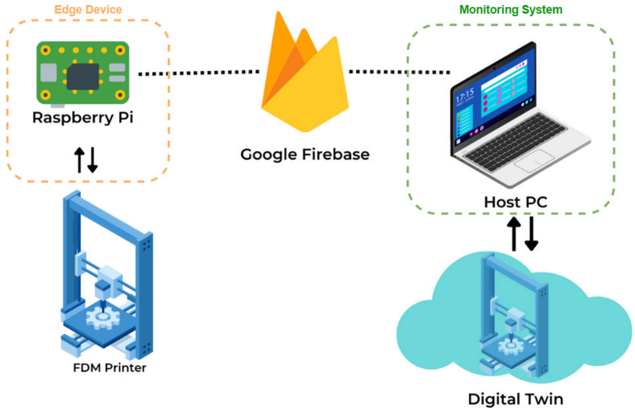

Figure 2. **The system architecture of the digital twin-based fault monitoring system is comprised of three primary components: the physical printer, the digital twin, and the monitoring system. The monitoring system utilizes data from both the physical printer and the digital twin to identify and diagnose faults in real-time.**

The utilization of the digital twin system provides notable benefits compared to solely relying on physical sensing information. Through the capture and analysis of supplementary data and parameters, the digital twin system enables a more extensive and precise comprehension of the behavior of the physical system. This augmented dataset empowers the training of machine learning models that possess an enhanced ability to understand and predict the dynamics of the system, resulting in improved performance, advanced monitoring capabilities, and well-informed decision-making.

B. MODEL DEVELOPMENT

The proposed digital twin-based fault monitoring system consists of three main components: (i) the physical 3D printer, (ii) the virtual printer, and (iii) the machine learning algorithms used for fault monitoring and behavior prediction. Figure 2 depicts the data flow between the printer and the digital twin platform proposed in this work, with the monitoring system situated on the host PC. The proposed monitoring system utilizes information from both physical and digital printers to assess the presence of faults.

The physical 3D printer is the actual machine that produces physical parts. It is equipped with embedded sensors that collect temperature data, which is periodically transmitted to the cloud server and stored in Google Firebase using the OctoPrint API installed on the Raspberry Pi connected to the printer.

The virtual printer is a digital replica of the physical printer. It simulates the printing process and behavior of the physical printer based on the data collected from the sensors. It obtains the data from Google Firebase and processes it on the host PC. Using machine learning algorithms, the digital twin can predict potential faults and diagnose any issues that may arise during printing.

The machine learning algorithms are responsible for analyzing the data collected from both the physical and virtual printers. The algorithms use this data to detect patterns and anomalies in the printing process that may indicate a fault. They can also predict potential faults and diagnose any issues that may arise during printing. Failure prediction algorithms act as fail-safe mechanisms that give operators time to mitigate errors by evaluating the operational conditions of a 3D printer and taking action accordingly [31]. In this research, the future time step of the sensor reading is predicted to anticipate any anomalies in machine operations. The sensors connected to the printer input measurement readings from time $t - 4$ to t, which are then processed using a regression model to output the prediction at time $t + 1$. In total, five historical data were used as input to the machine learning model. Below are the different regression models used and how they compare in predicting the future time step.

Choosing the appropriate machine learning model for a monitoring system can be a daunting task due to the large number of algorithms available. The process usually involves identifying the problem statement, analyzing the data, and choosing the most appropriate model that suits the system's requirements. In this case, the algorithms selected were ridge regression, XGBoost, Inception Time, Xception Time, Time Series Transformer (TST), Rocket Ridge, Logistic Regression, Rocket XGBoost, and ResNet.

The first step in choosing an appropriate machine learning model for a monitoring system is to understand the problem statement and the data. This involves collecting and preprocessing the data, identifying the variables, and defining the problem statement. Once this is done, the second step is to identify the most appropriate algorithm for the system's requirements.

The third step is to train and evaluate each of the models. This involves splitting the data into training and testing sets, training the models on the training set, and evaluating their performance on the testing set. Each model is evaluated based on accuracy, precision, recall, F1-score, and computation time. The top-performing models are then selected for ensembling. Based on the results discussed in the following subsections, the top four performing models are TST, Xception Time, ResNet, and Rocket+Ridge.

The design and modeling of a digital twin is critical in developing a fault monitoring system for additive manufacturing. A digital twin is a virtual representation of a physical object or system designed to mimic its behavior in the real world. In additive manufacturing, a digital twin can simulate the printing process and predict potential faults or defects [32]. This section provides an overview of the design and modeling process for a digital twin-based fault monitoring system, including the selection of modeling tools and the incorporation of real-world data. By leveraging the power of digital twins, manufacturers can improve their production processes and reduce the likelihood of faulty prints [33].

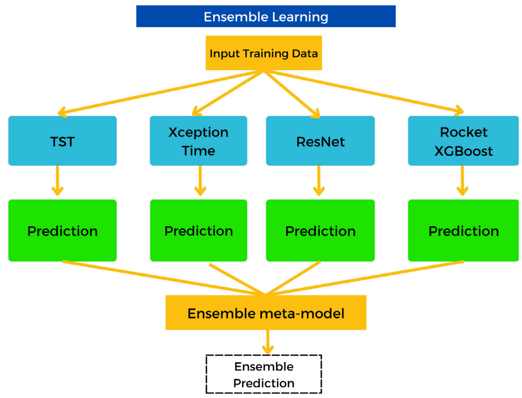

Figure 3. **The ensemble technique employed a stacking approach, where the predictions of the top four performing algorithms served as inputs to a meta-model. The meta-model was trained to generate the final output by leveraging the predictions of the base models. This ensembling method effectively enhanced the overall performance of the individual models and culminated in the creation of the 3D-AmplifAI algorithm.**

C. 3D-AMPLIFAI ALGORITHM

Various algorithms were employed to address a fault-monitoring problem, including Ridge regression, XG Boost, InceptionTime, TST, Rocket Ridge, Logistic Regression, Rocket XGBoost, ResNet, and Rocket Ridge regression. These algorithms were applied to identify mechanical faults in equipment, and their performances were compared based on metrics such as true positive, false negative, false positive, and true negative. Among these algorithms, TST, Xception Time, ResNet, and Rocket XGBoosting demonstrated superior performance and were combined to create a new algorithm called 3D-AmplifAI. The 3D-AmplifAI algorithm leveraged the strengths of these top-performing algorithms to enhance fault diagnosis accuracy.

To develop the ensemble algorithm, the four models were trained using a large dataset of temperature sensor data from an FDM printer. The performance of the 3D-AmplifAI algorithm was evaluated on a separate test dataset and surpassed the individual models, achieving higher accuracy and reducing false positives and false negatives. This highlights the potential of ensemble techniques in enhancing the performance of machine learning models for complex tasks like fault monitoring. Figure 3 illustrates the process of model ensembling in the development of the 3D-AmplifAI algorithm.

The overall pseudocode of the proposed 3D-AmplifAI is presented in Algorithm 1. Firstly, the dataset is loaded and preprocessed using Z-score and MinMaxScaler. It is then divided into three sets: testing, validation, and test. Secondly, the model is loaded, and a grid search is conducted for hyperparameter optimization. The best-performing model is selected and stored as the reference model for evaluating the test data.

Lastly, to ensure the reproducibility of the proposed 3D-AmplifAI, detailed configurations for each machine learning algorithm are provided as follows:

Algorithm 1 3D-AmplifAI Pseudocode

Input: Historical sensor data.
Output: Best ML model performance.
Initialize: ML model, list of epochs, and list of learning rate.
#Dataset Preparation.
Load dataset.
Calculate outlier using Z-score.
Normalize the dataset using MinMaxScaler.
Divide dataset into train:70%, val:10%, and test:20%.
#Model Initialization.
Model = Sequential ().
Model.add (TimeSeriesTransformerModel).
Model.add (Xception Time).
Model.add (ResNet).
Model.add (Rocket XGBoost).
Model.add (Dense).
#Model Training with Hyperparameter Tuning.
For epoch in epochs:
    For lr in learning rate:
        Train Model (epoch, lr) using train & val data.
        If result > best:
            Store result as best_conFigure
#Model Test.
Test Model (epoch, lr) using test data.
Calculate Accuracy, Precision, Recall, F1-Score, Computing Time, and Confusion Matrix.
Plot results using Matplotlib.

- TST is equipped with two layers, using a dropout rate of 0.1. It was trained for 100 epochs with a learning rate of 0.1.
- XceptionTime is designed with a kernel size of 50. It was trained for 100 epochs with a learning rate of 0.1.
- ResNet is trained with a depth of 18, referred to as ResNet18. It was trained for 100 epochs with a learning rate of 0.1.
- Rocket+XGBoost is initialized with a kernel size of 50 and a random state configuration of 111.

D. DIGITAL TWIN ENVIRONMENT

In developing the digital twin, a virtual environment for the 3D printer was designed. The digital twin design detailed in [34] was followed. The digital twin environment designed in this work consists of both physical and cyber spaces. Communication between the physical and virtual spaces is established using lightweight channels, specifically through the use of sockets. Unity software was used to model this environment and the printer laboratory itself. Unity is a popular game engine increasingly used in various fields, including digital twin development [35]. Unity offers several advantages that make it suitable for developing digital twin-based systems. For instance, Unity has a user-friendly interface that allows developers to create 3D models of systems, which can accurately simulate real-world behavior. Additionally, Unity has a rich library of resources, such as pre-built assets, textures, and shaders, which make the development process more efficient.

In the context of digital twin-based systems, Unity can create an interactive virtual environment that simulates the behavior of the physical system. The virtual environment is constructed using 3D models, textures, and animations that mimic the real-world scenario. Unity enables developers to create custom physics engines that accurately replicate the behavior of the physical system. This feature is particularly useful in digital twin-based systems as it allows users to test different scenarios and evaluate the impact of various factors on the behavior of the physical system. Another advantage of Unity in digital twin-based systems is its compatibility with various sensors and data acquisition devices. Unity supports several programming languages, such as C#, C++, and Java, which can interface with different hardware and software components. This enables developers to integrate sensors and data acquisition devices into the virtual environment, facilitating real-time monitoring and analysis of the physical system.

Moreover, Unity supports machine learning algorithms that can be used to develop predictive models forecasting the behavior of the physical system [36], [37]. Machine learning algorithms can be trained using historical data collected from the physical system, and the resulting models can be used to optimize the system's performance. In a digital twin-based system, machine learning models can predict faults and develop corrective actions that minimize downtime and improve efficiency.

In addition to modeling the physical components of the printer and its environment, it is also essential to consider the physics and mechanics involved in the 3D printing process. The behavior of the printer components, such as the extruder and print bed, must be accurately simulated to ensure that the virtual world behaves like the physical world. This can be achieved using physics engines and modeling software, such as Unity's built-in physics engine or external physics engines like Bullet or PhysX. Python is a programming language that can be used to create scripts to control and interact with the digital twin system, but it is not typically used for physics simulation in Unity.

Once the virtual world is created and the 3D printer is modeled, the next step is integrating the digital twin system with the physical 3D printer. This involves connecting the virtual world to the physical world through sensors and actuators. The sensors collect data from the physical printer, which is then used to update the virtual world. The virtual world can also control the physical printer by sending commands to the actuators. Figure 4 shows the environment of the virtual laboratory where the user may go around and inspect the different printers.

E. SENSOR DATA ACQUISITION AND INTEGRATION

Data collection from the 3D printer is a vital component of the fault monitoring system based on digital twins. Octoprint,

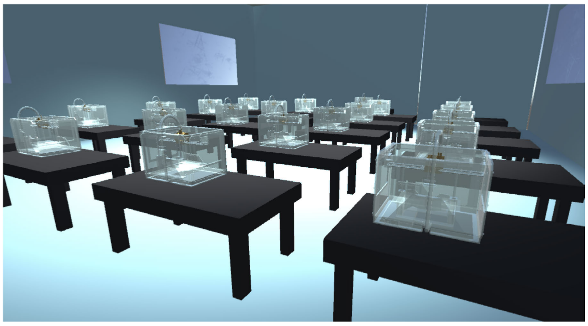

Figure 4. The virtual laboratory replicates a smart factory with various FDM printers that operate simultaneously.

an open-source program, monitors and controls 3D printers remotely. It allows real-time data gathering from the 3D printer's embedded sensors, such as the print temperature, bed temperature, print speed, and layer height. To collect data from the printer using Octoprint, the software is first installed on a Raspberry Pi. The Raspberry Pi is then connected to the 3D printer's control board via USB, allowing it to communicate with the printer and collect data on various parameters. By combining Octoprint with the device, real-time data can be gathered from the 3D printer and sent to the digital twin. This data is crucial for defect monitoring as it sheds light on the printer's actions during printing. Using this data, machine learning models can then be trained to find flaws and forecast future behavior.

Integrating the physical and virtual worlds is a crucial aspect of digital twin technology. It connects physical assets, such as machines or equipment, to virtual counterparts in a digital twin environment. This integration enables the digital twin to receive real-time data from the physical asset and use it to create a dynamic model that represents the current state of the asset. The integration process is accomplished by using sensors attached to the physical asset and collecting various data types, such as temperature, pressure, speed, and vibration. The collected data is transmitted to the digital twin environment through a communication network. Ensuring that the digital twin environment accurately reflects the physical asset is essential to achieve seamless integration between the physical and virtual worlds. This requires a thorough understanding of the physical asset and its behavior and the ability to model it accurately in the digital twin environment.

F. DATA PREPROCESSING

Preprocessing the data from Octoprint is essential in building an effective fault monitoring system for the 3D printer. The collected data is often raw and unorganized and requires processing to extract meaningful features and make it suitable for analysis. Preprocessing steps typically include data cleaning, normalization, and feature extraction.

Data cleaning involves identifying and handling missing, erroneous and removing any noise or outliers that may impact the accuracy of the analysis. In 3D printing, this could include removing data points outside the normal range of printer operations. In this work, we utilize the Z-score to determine outliers, which is calculated using Equation (1):

$$Z = \frac{x - \mu}{\sigma} \tag{1}$$

where x represents the data point, and μ and σ represent the mean and standard deviation of the data, respectively. If the Z-score equals or exceeds 2, the corresponding data point is excluded from the dataset.

Initially, the dataset contained a total of 56,750 instances before undergoing preprocessing. Following the data clearing process, 56,100 instances were filtered and used for training, validation, and testing the model. The data were then normalized using MinMax scaling, which can be calculated using Equation (2):

$$x_{norm} = \frac{x - min(x)}{max(x) - min(x)} \tag{2}$$

in this context, x_{norm} denotes the normalized value of the data, while x represents the current value. $min(x)$ and $max(x)$ refer to the minimum and maximum values of x in the dataset, respectively. Through the MinMax normalization process, the range of all features is adjusted to span from 0 to 1. The dataset was partitioned with 70% of instances used for training, 10% for validation, and 20% for testing purposes.

Furthermore, normalization involves scaling the data to ensure all variables are on a similar scale, which is essential when using machine learning algorithms to analyze the data. Feature extraction involves identifying the relevant features or attributes from the data that used for analysis. In the context of 3D printing, these could include print speed, temperature, filament usage, and other printer settings. Feature extraction is crucial because it reduces the complexity of the data, making it easier to analyze and more efficient for machine learning algorithms to process.

G. DATA TRANSMISSION

A cloud-based platform called Google Firebase provides tools for building and growing mobile and online applications. It enables the creation of real-time apps, data synchronization across numerous devices, and user authentication using social media or email services.

Google Firebase can be used as a data-passing platform in the context of a digital twin system for 3D printing to transfer data from the actual 3D printer to the virtual environment made in Unity. On the Raspberry Pi linked to the 3D printer, the Octoprint plugin can be installed to accomplish this. The plugin enables the communication between the Raspberry Pi and OctoPrint and data transmission to the Firebase Real-time Database.

Google Firebase keeps synchronized updates between connected clients in real-time and saves data in JSON format. Once the information is transferred to the Firebase Real-time Database, it is accessible from any location with an internet connection, including the Unity-created virtual world.

A script can be created to periodically get information from the Firebase Real-time Database and update the virtual 3D

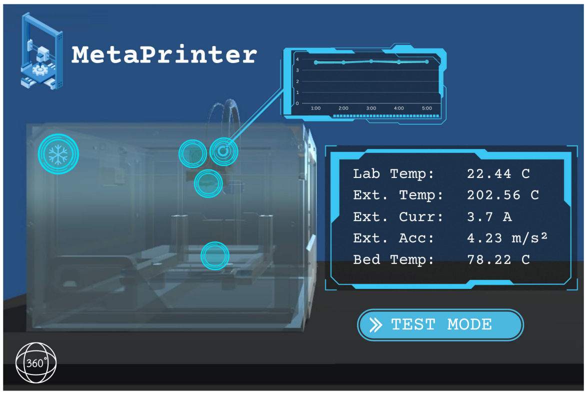

Figure 5. The 3D printers may be accessed and the different parameters of the printer are displayed.

printer model accordingly in the virtual environment. For instance, if the extrusion temperature in the real-world 3D printer grows, the extruder temperature in Unity will similarly rise in real-time, accurately simulating the real-world printer in the virtual one.

Firebase can be used to transfer information the other way, from the virtual setting to the physical printer. For instance, if the virtual environment notices a problem or error while printing, it can send a notification to the Firebase Real-time Database, which then causes the actual printer to take some action, like halting printing or modifying the extruder's temperature.

IV. PREDICTION AND SIMULATION

Prediction and simulation are crucial components of a digital twin-based system. The ability to forecast future behavior and simulate potential outcomes can aid in decision-making processes and enhance system performance. This section detailed the techniques and methods used for prediction and simulation in the context of fault monitoring in additive manufacturing using a digital twin-based system. The use of machine learning algorithms and simulation models to predict future behavior and simulate potential outcomes of the physical system was explained. The results of these predictions and simulations can then be used to inform decisions and optimize the system's performance.

A. BEHAVIOUR PREDICTION

In a digital twin system, machine learning algorithms can be used to predict the future behavior of a 3D printer. This involves collecting and preprocessing data from the physical printer using technologies such as OctoPrint and Firebase, as discussed in previous sections.

As part of the system development, historical data is fed into machine learning algorithms to enable them to recognize patterns and correlations in the data, which helps in predicting future behavior and detecting and classifying faults. Figure 5 shows how a 3D printer within the virtual laboratory may be inspected.

Table 1. The 3D printer experimental conditions.

3D Printer	Creality Ender-5
Type	Fused Deposition Modelling
Filament Material	ABS, PLA, PVA, Hips, Nylon, PC, Flex, Petg, Rubber, SBS, PP
Filament Diameter	1.75 mm
Nozzle Diameter	1.75 mm
Speed	80 mm/s
Nozzle Temperature	260 °C
Hotbed Temperature	100 °C
Area Size	220 x 220 x 300

The Python-based programming language is used to write scripts that process data collected from the 3D printer through OctoPrint and to train machine learning models to predict future behavior. These scripts are then imported into Unity as custom assets and used to create the logic that drives the digital twin simulation.

B. FAULT DIAGNOSIS

One way that Octoprint aids in fault mitigation is by providing real-time information about the printer's temperature, extrusion rate, and other parameters. This information can be used to identify issues such as clogs in the extruder or inconsistencies in the temperature, which can then be corrected. For example, if the extruder's temperature is too high, Octoprint can reduce the temperature to the appropriate level.

C. FAULT MITIGATION

Fault mitigation is a critical aspect of 3D printing as it involves identifying and resolving errors or issues that may arise during printing [38]. Once anomalies in the predicted temperature readings of the printer occur, controlling the 3D printer with Octoprint can aid in fault mitigation. The printing process can be stopped immediately, and the issue can be addressed. This can be particularly useful when the printer is located in a different room or building, as it allows for immediate action to be taken without having to access the printer physically.

Controlling the 3D printer with Octoprint can mitigate fault by providing real-time information, additional functionality through plugins, and remote control capabilities. Using Octoprint, errors can be identified and corrected quickly, ensuring that printing is completed successfully and with minimal issues.

V. EXPERIMENTAL SETUP

The experimental setup for the proposed system was applied to a Creality Ender 5 FDM printer, with the parameters displayed in Table 1. The proposed approach is divided into two parts: monitoring and prognosis. For in-situ monitoring, an embedded temperature sensor collects the nozzle temperature values (T_e) and serves as input data for the machine

Figure 6. Sample prints of the STL file (left) produced both high-quality (center) and faulty output (right).

learning model. The proposed schemes are responsible for predicting sensor values at a future timestep $t+1$ and detecting any possible anomalies that may occur. By anticipating any abnormalities, the system can prepare and avoid potential errors.

The 3D printer's embedded sensors were used for printer monitoring. These sensors were accessed using a Raspberry Pi running the OctoPi operating system that runs OctoPrint. The experimental setup was designed to collect data from the printer's sensors and develop a machine learning algorithm that can accurately predict normal and abnormal operating conditions. The collected data was used to train and test the machine learning algorithm developed using Python and TensorFlow. The algorithm utilized the collected sensor data to predict potential failures in the printing process, such as nozzle clogs or filament jams, with high accuracy.

To monitor the behavior of a 3D printer, data is collected from the printer during a printing job. This data is collected every 1000 milliseconds and includes various parameters such as extruder temperature, bed temperature, and environment temperature. The extruder temperature indicates the temperature of the printer's hot end, which is responsible for melting the plastic filament used to create the 3D object. The bed temperature refers to the temperature of the printing bed, which is essential for ensuring proper adhesion of the printed object to the bed. The environment temperature refers to the temperature of the surrounding environment, which can impact the printer's performance. It is worth mentioning that collecting data every 1000 milliseconds allows for the detection of changes that may occur on shorter timescales. Furthermore, collecting data over multiple printing jobs provides a more comprehensive view of the printer's behavior over time, which enables more accurate predictions of future behavior.

A test file was printed 114 times to generate data, with 53 prints resulting in faulty output labeled as "error" and 61 prints of good quality labeled as "fine". To simulate the errors produced by faulty printers, the GCode of the 3D model was created to force errors using a randomizer that allowed printer parameters to fluctuate and replicate scenarios where the printer malfunctions. Error prints include blowouts, separation, gaps between layers, and stringy. On the other hand, quality prints are those with minimal to no surface issues. Figure 6 shows what the STL file looks like, as well as an example of a quality print and a defective print.

Table 2. Digital twin experimental implementation details.

System Parameter	Details
Operating System	Windows 10
Unity Environment	Unity Game Engine
3D Printer Asset Design	Blender
ML Models	TensorFlow
Communication	SocketIO

Another significant aspect of digital twin development, as discussed in this article, is the implementation settings of the virtual environment. Table 2 provides detailed information on the overall configuration used to design the digital twin for 3D fault detection in this study. The digital twin system was developed by dividing physical and cyber space, following the concept of a digital twin manufacturing cell. The digital twin environment is designed using the Unity game engine within a Windows 10 environment. To facilitate 3D modeling of the printer, Blender was utilized to create and export the model. The exported model was then incorporated into Unity to design a laboratory environment, as depicted in Figure 4. The virtual printer was designed to mimic the functionality of the physical printer and also includes the implementation of a machine learning model.

Regarding data transmission requirements within the digital twin, a low-overhead communication channel called SocketIO was employed for efficient communication between the server and the client. With this design, the digital twin of the physical 3D printer has been established and can be utilized for print simulation purposes. It is worth mentioning that the design of the digital twin can be implemented in smart manufacturing, especially for FDM 3D printer configurations, as the 3D and machine learning models are specifically designed for the plastic printer.

VI. PERFORMANCE EVALUATION

This section covers the model hyperparameter tuning to obtain the best parameter, specifically the learning rate for the proposed model. The best-performing learning rate was used to compare the performance of other machine learning-based approaches for fault prediction in 3D printers for additive manufacturing.

A. MODEL HYPERPARAMETERS

To get the optimum performance, a number of hyperparameters are tuned and optimized during the training of deep learning models. The learning rate, which regulates the magnitude of the updates to the model's parameters during training, is one of the essential hyperparameters. If the learning rate is too high, the model could overshoot the ideal values and produce unstable training. However, a low learning rate might lead to slower convergence and more extended training periods.

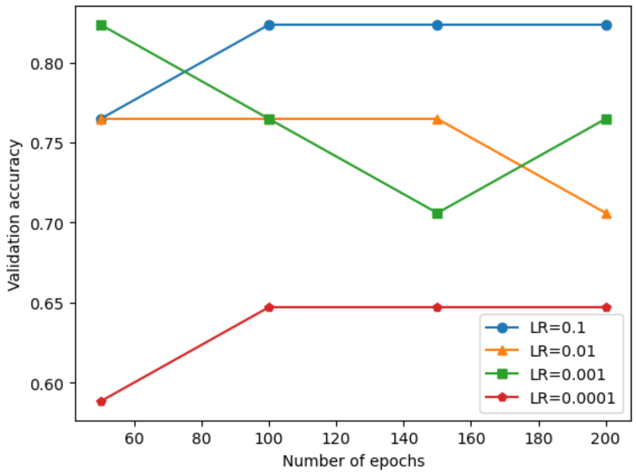

Figure 7. The validation accuracy of the 3D-AmplifAI model is evaluated across different learning rates and epochs.

Another crucial hyperparameter to consider is the number of epochs, representing the iterations of the model over the complete training dataset. The influence of epochs on overfitting is contingent upon several factors, including the complexity of the model, the size and diversity of the dataset, and the utilization of regularization techniques. It is essential to regulate the number of epochs appropriately to avoid the risk of overfitting, where the model memorizes the training data rather than acquiring general patterns applicable to new data.

After performing a grid search, it was found that the best-performing models were obtained with a learning rate of 0.1 and 100 epochs, as shown in Figure 7. However, it is important to note that the optimal hyperparameters may vary depending on the specific dataset and problem being addressed. Therefore, it is recommended to conduct a comprehensive hyperparameter tuning process for each new dataset in order to achieve optimal results.

B. COMPARATIVE ANALYSIS

After obtaining the best hyperparameter configuration, those settings are used to evaluate the proposed ensemble model and various machine learning algorithms. The test data is used to evaluate the accuracy of each model. Figure 8 illustrates the performance of the evaluated models ranging from 0 to 1. The closer the accuracy is to 1, the better the model's performance in predicting the 3D printer fault condition. The highest-performing model was obtained using the proposed 3D-AmplifAI model with an accuracy of 0.8235. Xception Time, XGBoost, and TST were also able to achieve an accuracy of 0.7647. In addition, the lowest performance was produced by the Rocket+XGBoost model with an accuracy of 0.667. Based on these results, the proposed model is able to enhance the fault detection performance up to 7.689% due to its capability to combine multiple results from different machine learning models using an ensemble approach.

The next performance metric evaluated in this paper is the confusion matrix that can be used to evaluate the fault

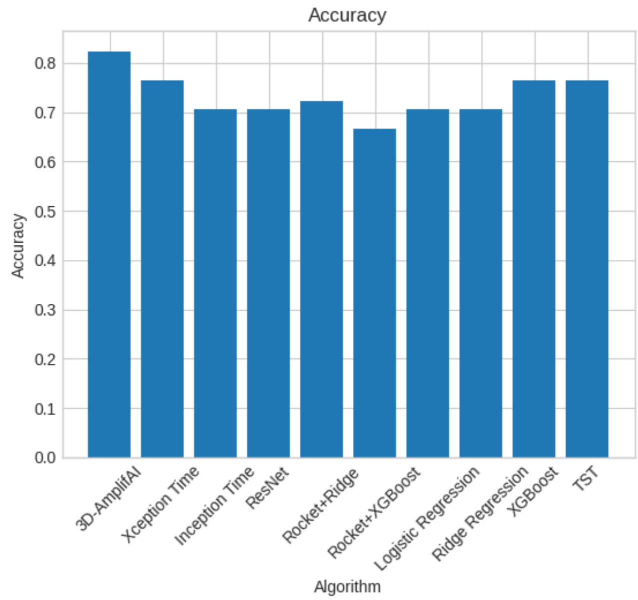

Figure 8. The comparison of various machine learning models in terms of accuracy using test data.

classification model that has been constructed. These categories are true positive (TP), true negative (TN), false positive (FP), and false negative (FN) in the matrix. The number of positive cases the model correctly predicts is the TP. (i.e., faults correctly identified by the model). The number of negative occurrences the model correctly predicts is TN. (i.e., non-faults correctly identified by the model). FPs are occurrences of non-faults that are mistakenly categorized as faults. The number of faults mistakenly labeled as non-faults is known as an FN.

The generated model's efficiency at correctly identifying defects can be assessed using performance metrics like precision, recall, accuracy, and F1-score, which can be calculated using the confusion matrix. Precision is the proportion of actual positive instances to all expected positive ones. The recall is the proportion of real positives to all real positive cases. The proportion of occurrences that were correctly classified to all of the instances is called accuracy. The harmonic mean of recall and precision is the F1-score.

The developed model's effectiveness in accurately identifying faults can be determined by analyzing the confusion matrix and calculating these performance metrics. The results can also help identify areas for improvement in the model development process.

Table 3 display the different values obtained after testing the dataset on all machine learning models and Figure 9 illustrates the confusion matrix of the proposed 3D-AmplifAI model. Based on the results, the model with the highest number of TPs is Xception Time, with 4,620. This means that the model correctly identified 4,620 instances of mechanical faults. The model with the second-highest number of TPs is TST, with 3,300.

On the other hand, the model with the lowest number of TPs is Ridge Regression, with only 1,980. This means the

Table 3. Confusion matrix summary.

Model	True Positive	False Negative	False Positive	True Negative
Xception Time	4,620	3,960	1,980	660
Inception Time	3,300	4,620	1,320	1,980
ResNet	3,960	3,960	1,980	1,320
Rocket+Ridge	2,640	5,940	660	2,640
Rocket XGBoost	3,300	4,620	1,980	1,980
Logistic Regression	2,640	5,280	1,980	1,320
Ridge Regression	1,980	5,940	1,320	1,980
XGBoost	2,640	5,940	1,320	1,980
TST	3,300	5,280	660	1,980
3D - AmplifAI (Proposed)	**3,960**	**5,280**	**660**	**1,320**

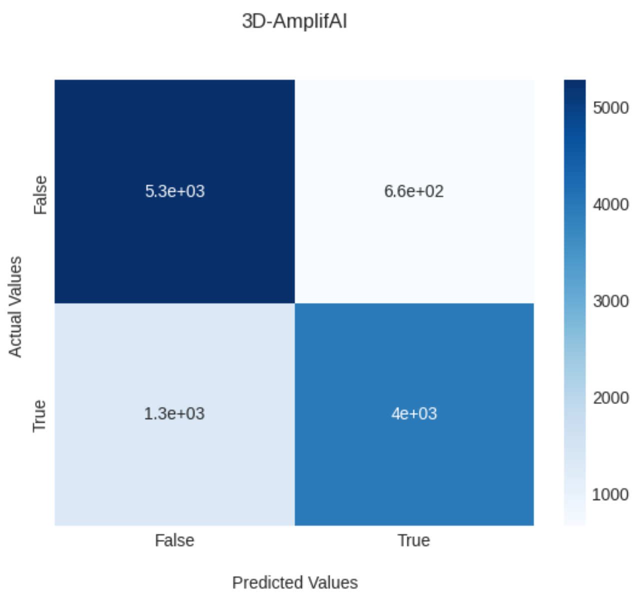

Figure 9. Confusion matrix of the proposed 3D-AmplifAI ensemble model.

model identified only 1,980 mechanical faults, much lower than the other models. In terms of FNs, the model with the lowest number is ResNet, which only missed 3,960 instances of mechanical faults. In contrast, the model with the highest number of FNs is Rocket Ridge, which missed 5,940 instances. The model with the lowest number of FPs is Rocket Ridge, with only 660. This means that the model wrongly identified only 660 instances as mechanical faults. In contrast, the model with the highest number of FPs is Inception Time with 1980. Overall, the Xception Time model seems the best-performing model based on the highest number of TPs. However, further analysis is required to determine which model is the most appropriate for this specific use case.

Several factors may contribute to the values in the table. First, the precision, accuracy, recall, and F1-score are all measures of the performance of the various models used in the study. These metrics have a range from 0 to 1, where the closest performance to 1 indicates the better model performance. These metrics are commonly used to evaluate the performance of machine learning models in classification problems. Therefore, the detailed performance of each machine learning model is shown in Table 4.

The 3D-AmplifAI model performed the best in precision, accuracy, recall, and F1- score, achieving scores of 0.8571, 0.8235, 0.7500, and 0.8000, respectively. This may be due to the fact that it is an ensemble model combining the strengths of multiple algorithms (TST, Xception Time, ResNet, and Rocket XGBoost) to achieve better performance. Additionally, it had the longest training time of all the models, suggesting that it could learn more complex patterns in the data.

Xception Time and Inception Time models also performed relatively well in accuracy, recall, and F1- score, achieving scores above 0.7 for each metric. These models are based on deep learning architectures that are specifically designed for time series data, which contributed to their good performance on this problem. Rocket+Ridge, Rocket+XGBoost, Logistic Regression, and Ridge Regression models performed relatively poorly compared to the other models, achieving F1-scores below 0.7. These models are all based on linear or logistic regression algorithms, which may not be able to capture the complex patterns present in the data as effectively as the other models.

Finally, it is worth noting that the computation time for each model varies widely, with the fastest model (XGBoost) taking only 0.0034 seconds to compute. In comparison, the slowest model (3D-AmplifAI) took over 200 times longer at 0.2155 seconds. The choice of the model may also depend on the computational resources available and the speed at which predictions need to be made.

C. MODEL EFFICIENCY ANALYSIS

The final performance metric investigated in this article is model efficiency. In this case, the computing time required to perform the prediction is calculated. To achieve this, a timestamp is set before the prediction process is conducted. After the prediction process is completed, the new timestamp is determined, and the time difference between the two timestamps is divided by the total prediction samples to represent the processing time of each model to generate a single prediction. Figure 10 shows the various computation times of the proposed 3D-AmplifAI model and other machine learning models. The results indicate that the proposed ensemble model achieves the longest prediction time among the other models, with 215.5ms. On the other hand,

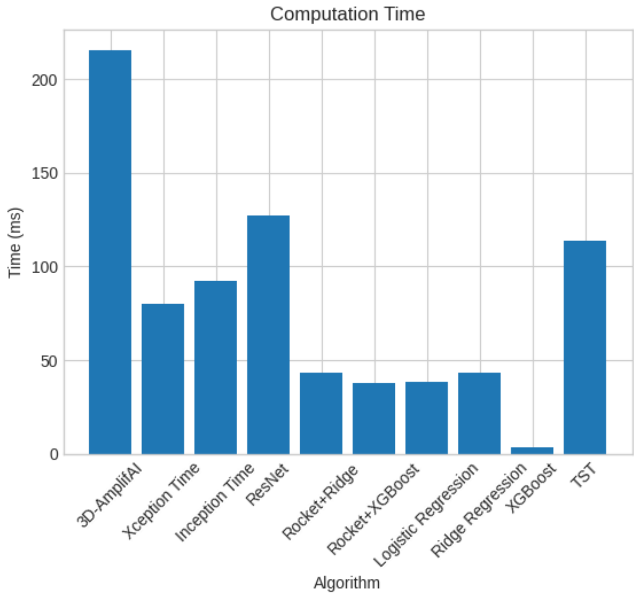

Figure 10. **The comparison of various machine learning models in terms of computing time to generate a single prediction.**

Table 4. **Summary of precision, recall and F1-score from various machine learning algorithms investigated in this work.**

Model	Precision	Recall	F1- Score
Xception Time	0.7000	0.8750	0.7778
Inception Time	0.7143	0.6250	0.6667
ResNet	0.6667	0.7500	0.7059
Rocket+Ridge	0.8000	0.5000	0.6154
Rocket+XGBoost	0.6250	0.6250	0.6250
Logistic Regression	0.5714	0.6667	0.6154
Ridge Regression	0.6000	0.5000	0.5455
XGBoost	0.6667	0.6667	0.6667
TST	0.8333	0.6250	0.6250
3D - AmplifAI (Proposed)	**0.8571**	**0.7500**	**0.8000**

the XGBoost model is the fastest in conducting the prediction, with a processing time of 3.4ms, which is significantly fast. However, when comparing the trade-off between XGBoost and the proposed 3D-AmplifAI in terms of model efficiency and performance, the proposed ensemble model is able to achieve significantly better performance compared to XGBoost. Additionally, the computing time of 215.5ms is still considerably fast to detect any faults in a real-time system.

VII. CONCLUSION AND FUTURE WORKS

The study developed a digital twin-based system for monitoring faults in a 3D printer using the 3D-AmplifAI algorithm. The results showed that the 3D-AmplifAI algorithm had the highest accuracy (82.35%), precision (85.71%), and F1-score (80%) among the nine algorithms tested, making it a practical approach for fault monitoring in 3D printing applications. Compared to the other algorithms, including XGBoost and Rocket Ridge, the 3D-AmplifAI algorithm demonstrated superior accuracy, precision, and F1-score values. Also, the performance trade-off between the proposed 3D-AmplifAI is better compared to other machine learning-based models presented in Section VI. The study suggests that integrating this system into a 3D printer could improve its efficiency and reliability.

The recommendations for future work include incorporating more sensors to capture additional variables, integrating other machine learning techniques, developing real-time decision-making capabilities, expanding the system to cover other additive manufacturing processes, deploying the system on the cloud for remote access and monitoring, incorporating feedback mechanisms, and collaborating with industry partners to validate the system's effectiveness in real-world scenarios. These future work suggestions aim to improve the system's accuracy, versatility, and applicability in various manufacturing industries and enhance its effectiveness in mitigating faults. Additionally, applying a federated learning approach could potentially reduce the communication cost required for updating the machine learning model, particularly when combined with an effective client selection mechanism [39].

REFERENCES

[1] A. Haleem and M. Javaid, "Additive manufacturing applications in industry 4.0: A review," *J. Ind. Integr. Manag.*, vol. 4, no. 4, 2019, Art. no. 1930001.

[2] M. Javaid and A. Haleem, "Additive manufacturing applications in medical cases: A literature based review," *Alexandria J. Med.*, vol. 54, no. 4, pp. 411–422, Dec. 2018.

[3] S. M. E. Sepasgozar, A. Shi, L. Yang, S. Shirowzhan, and D. J. Edwards, "Additive manufacturing applications for industry 4.0: A systematic critical review," *Buildings*, vol. 10, no. 12, p. 231, Dec. 2020.

[4] Grand View Research, "3D printing market size, share & trends analysis report by component (hardware, software, services), by printer type, by technology, by software, by application, by vertical, by region, and segment forecasts," San Francisco, CA, USA, Market Anal. Rep. 978-1-68038-000-2, 2020, p. 250.

[5] P. Charalampous, I. Kostavelis, C. Kopsacheilis, and D. Tzovaras, "Vision-based real-time monitoring of extrusion additive manufacturing processes for automatic manufacturing error detection," *Int. J. Adv. Manuf. Technol.*, vol. 115, nos. 11–12, pp. 3859–3872, Aug. 2021.

[6] A. Malik, M. I. U. Haq, A. Raina, and K. Gupta, "3D printing towards implementing industry 4.0: Sustainability aspects, barriers and challenges," *Ind. Robot, Int. J. Robot. Res. Appl.*, vol. 49, no. 3, pp. 491–511, Apr. 2022.

[7] G. A. R. Sampedro, S. M. Rachmawati, D.-S. Kim, and J.-M. Lee, "Exploring machine learning-based fault monitoring for polymer-based additive manufacturing: Challenges and opportunities," *Sensors*, vol. 22, no. 23, p. 9446, Dec. 2022.

[8] G. D. Goh, S. L. Sing, and W. Y. Yeong, "A review on machine learning in 3D printing: Applications, potential, and challenges," *Artif. Intell. Rev.*, vol. 54, no. 1, pp. 63–94, Jan. 2021.

[9] C. Wang, X. P. Tan, S. B. Tor, and C. S. Lim, "Machine learning in additive manufacturing: State-of-the-art and perspectives," *Additive Manuf.*, vol. 36, Dec. 2020, Art. no. 101538.

[10] S. Y. Korostelev, E. E. Slyadnikov, and I. Y. Turchanovsky, "The physical foundations of 3D-printing technology. Molecular dynamics simulation," *Russian Phys. J.*, vol. 65, pp. 1–9, Jan. 2023.

[11] A. Mihalache, A. Hriţuc, M. Boca, B. Oroian, I. Condrea, C. Botezatu, and L. Slătineanu, "Thermal insulation capacity of a 3D printed material," *Macromolecular Symposia*, vol. 396, no. 1, Apr. 2021, Art. no. 2000286.

[12] M. Liu, S. Fang, H. Dong, and C. Xu, "Review of digital twin about concepts, technologies, and industrial applications," *J. Manuf. Syst.*, vol. 58, pp. 346–361, Jan. 2021.

[13] C. Qian, X. Liu, C. Ripley, M. Qian, F. Liang, and W. Yu, "Digital twin—Cyber replica of physical things: Architecture, applications and future research directions," *Future Internet*, vol. 14, no. 2, p. 64, Feb. 2022.

[14] L. Zhang, X. Chen, W. Zhou, T. Cheng, L. Chen, Z. Guo, B. Han, and L. Lu, "Digital twins for additive manufacturing: A state-of-the-art review," *Appl. Sci.*, vol. 10, no. 23, p. 8350, Nov. 2020.
[15] Y. Jiang, S. Yin, K. Li, H. Luo, and O. Kaynak, "Industrial applications of digital twins," *Philos. Trans. Roy. Soc. A*, vol. 379, no. 2207, 2021, Art. no. 20200360.
[16] T. Kalsoom, N. Ramzan, S. Ahmed, and M. Ur-Rehman, "Advances in sensor technologies in the era of smart factory and industry 4.0," *Sensors*, vol. 20, no. 23, p. 6783, Nov. 2020.
[17] E. Tzimas, G.-C. Vosniakos, and E. Matsas, "Machine tool setup instructions in the smart factory using augmented reality: A system construction perspective," *Int. J. Interact. Design Manuf.*, vol. 13, no. 1, pp. 121–136, Mar. 2018.
[18] F. Longo, L. Nicoletti, and A. Padovano, "Ubiquitous knowledge empowers the smart factory: The impacts of a service-oriented digital twin on enterprises' performance," *Annu. Rev. Control*, vol. 47, pp. 221–236, Jan. 2019.
[19] R. Ashima, A. Haleem, S. Bahl, M. Javaid, S. K. Mahla, and S. Singh, "Automation and manufacturing of smart materials in additive manufacturing technologies using Internet of Things towards the adoption of industry 4.0," *Mater. Today*, vol. 45, pp. 5081–5088, Jan. 2021.
[20] D. J. S. Agron, J.-M. Lee, and D.-S. Kim, "Nozzle thermal estimation for fused filament fabricating 3D printer using temporal convolutional neural networks," *Appl. Sci.*, vol. 11, no. 14, p. 6424, Jul. 2021.
[21] S. Zhang, Z. Sun, C. Li, D. Cabrera, J. Long, and Y. Bai, "Deep hybrid state network with feature reinforcement for intelligent fault diagnosis of delta 3-D printers," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 779–789, Feb. 2020.
[22] S. Belikovetsky, Y. A. Solewicz, M. Yampolskiy, J. Toh, and Y. Elovici, "Digital audio signature for 3D printing integrity," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1127–1141, May 2019.
[23] C. Liu, A. C. C. Law, D. Roberson, and Z. Kong, "Image analysis-based closed loop quality control for additive manufacturing with fused filament fabrication," *J. Manuf. Syst.*, vol. 51, pp. 75–86, Apr. 2019.
[24] M. Putra, A. Chijioke, M. Verana, D.-S. Kim, and J.-M. Lee, "Efficient 3D printer fault classification using a multi-block 2D-convolutional neural network," *J. Korean Inst. Commun. Inf. Sci.*, vol. 47, no. 2, pp. 236–245, Feb. 2022.
[25] D. Chowdhury, A. Sinha, and D. Das, "XAI-3DP: Diagnosis and understanding faults of 3-D printer with explainable ensemble AI," *IEEE Sensors Lett.*, vol. 7, no. 1, pp. 1–4, Jan. 2023.
[26] G. L. Knapp, T. Mukherjee, J. S. Zuback, H. L. Wei, T. A. Palmer, A. De, and T. DebRoy, "Building blocks for a digital twin of additive manufacturing," *Acta Mater.*, vol. 135, pp. 390–399, Aug. 2017.
[27] T. DebRoy, W. Zhang, J. Turner, and S. S. Babu, "Building digital twins of 3D printing machines," *Scripta Mater.*, vol. 135, pp. 119–124, Jul. 2017.
[28] Y. G. Kabaldin, P. V. Kolchin, D. A. Shatagin, M. S. Anosov, and A. A. Chursin, "Digital twin for 3D printing on CNC machines," *Russian Eng. Res.*, vol. 39, no. 10, pp. 848–851, Oct. 2019.
[29] P. Stavropoulos, A. Papacharalampopoulos, C. K. Michail, and G. Chryssolouris, "Robust additive manufacturing performance through a control oriented digital twin," *Metals*, vol. 11, no. 5, p. 708, Apr. 2021.
[30] M. P. Ciano, R. Pozzi, T. Rossi, and F. Strozzi, "Digital twin-enabled smart industrial systems: A bibliometric review," *Int. J. Comput. Integr. Manuf.*, vol. 34, nos. 7–8, pp. 690–708, Aug. 2021.
[31] R. J. M. Wolfs, F. P. Bos, and T. A. M. Salet, "Hardened properties of 3D printed concrete: The influence of process parameters on interlayer adhesion," *Cement Concrete Res.*, vol. 119, pp. 132–140, May 2019.
[32] H. Mu, F. He, L. Yuan, P. Commins, H. Wang, and Z. Pan, "Toward a smart wire arc additive manufacturing system: A review on current developments and a framework of digital twin," *J. Manuf. Syst.*, vol. 67, pp. 174–189, Apr. 2023.
[33] T. N. Nguyen, R. Ponciroli, P. Bruck, T. C. Esselman, J. A. Rigatti, and R. B. Vilim, "A digital twin approach to system-level fault detection and diagnosis for improved equipment health monitoring," *Ann. Nucl. Energy*, vol. 170, Jun. 2022, Art. no. 109002.
[34] J. Zhang, P. Li, and L. Luo, "Digital twin-based smart manufacturing cell: Application case, system architecture and implementation," *J. Phys., Conf. Ser.*, vol. 1884, no. 1, Apr. 2021, Art. no. 012017.
[35] S. Mihai, W. Davis, D. V. Hung, R. Trestian, M. Karamanoglu, B. Barn, R. V. Prasad, H. Venkataraman, and H. X. Nguyen, "A digital twin framework for predictive maintenance in industry 4.0," in *Proc. Int. Conf. High Perform. Comput. Simulation*, Mar. 2021.
[36] N.-S. Pai, P.-X. Chen, P.-Y. Chen, and Z.-W. Wang, "Home fitness and rehabilitation support system implemented by combining deep images and machine learning using unity game engine," *Sens. Mater*, vol. 34, pp. 1971–1990, May 2022.
[37] C. G. Diaz, P. Perry, and R. Fiebrink, "Interactive machine learning for more expressive game interactions," in *Proc. IEEE Conf. Games (CoG)*, Aug. 2019, pp. 1–2.
[38] Q. Fang, G. Xiong, M. Zhou, T. S. Tamir, C. Yan, H. Wu, Z. Shen, and F. Wang, "Process monitoring, diagnosis and control of additive manufacturing," *IEEE Trans. Autom. Sci. Eng.*, early access, Nov. 14, 2022, doi: 10.1109/TASE.2022.3215258.
[39] M. A. P. Putra, A. R. Putri, A. Zainudin, D.-S. Kim, and J.-M. Lee, "ACS: Accuracy-based client selection mechanism for federated industrial IoT," *Internet Things*, vol. 21, Apr. 2023, Art. no. 100657.

GABRIEL AVELINO R. SAMPEDRO (Member, IEEE) received the B.S. and M.S. degrees in computer engineering from Mapúa University, Manila, Philippines, in 2018, and the Ph.D. degree in IT convergence engineering from the Kumoh National Institute of Technology, Gumi, South Korea, in 2023. Currently, he is an Assistant Professor at the Faculty of Information and Communication Studies at the University of the Philippines - Open University and a researcher at the Center for Computational Imaging and Visual Innovations at the De La Salle University. His research interests include real-time systems, embedded systems, robotics, and biomedical engineering.

MADE ADI PARAMARTHA PUTRA received the M.S. degree in electrical and computer engineering from the Bandung Institute of Technology, West Java, Indonesia, in 2019. He is currently a full-time Lecturer in informatics engineering with STMIK Primakara, Bali, Indonesia. His main research interests include named data networks (NDN), the real-time Internet of Things, federated learning optimization, and energy efficient architecture.

MIDETH ABISADO (Member, IEEE) is currently an Associate Member of the National Research Council of the Philippines and a Board Member of the Computing Society of the Philippines Special Interest Group for Women in Computing. She is also the Director of the CCIT Graduate Programs Department, National University, Manila. She heads research on harnessing natural language processing for community participation. social science, machine learning, and natural language processing principles and techniques are used in the study. It is well anticipated that thematic based on dashboard analytics will be used for policy recommendations for the government. She has 23 years of experience in education and research. Her passion is to encourage women to choose careers in computing and prepare and mold the next generation of Filipino IT professionals and leaders in the country. Her research interests include emphatic computing, social computing, human–computer interaction, and human language technology.