“SEAMSENSE” REAL-TIME QUALITY MONITORING SYSTEM FOR THE APPAREL INDUSTRY

Samaraweera G.P.M.D

R24-066

B.Sc. (Hons) Degree in Information Technology Specializing in Data Science

Department of Computer Science

Sri Lanka Institute of Information Technology
Sri Lanka

September 2024

“SEAMSENSE” REAL-TIME QUALITY MONITORING SYSTEM FOR THE APPAREL INDUSTRY

Samaraweera G.P.M.D – IT21016066

R24-066

Dissertation submitted in partial fulfillment of the requirements for the Bachelor of Science (Hons) in Information Technology Specializing in Data Science

Department of Computer Science

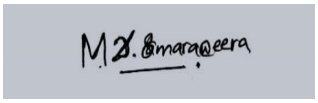
Sri Lanka Institute of Information Technology
Sri Lanka

September 2024

DECLARATION

I declare that this is our own work, and this dissertation does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other University or institute of higher learning, and to the best of our knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Also, I hereby grant to the Sri Lanka Institute of Information Technology, the non-exclusive right to reproduce and distribute our dissertation, in whole or in part in print, electronic or other mediums. We retain the right to use this content in whole or part in future works (such as articles or books).

Name	Student ID	Signature
Samaraweera G. P. M. D.	IT21016066	

Date – 08/22/2024

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Name	Signature
Mr. Samadhi Chathuranga Rathnayaka (Supervisor)	

Date – 08/22/2024

ACKNOWLEDGEMENT

For their invaluable advice and assistance during the research process, I would like to sincerely thank my dissertation supervisor, Mr. Samadhi Rathnayake, and co-supervisor, Ms. Supipi Virajini Karunathilake of the Faculty of Computing. Their insightful remarks and beneficial criticism helped me to improve the overall quality of my work and refine my study.

I extend my heartfelt thanks to the dedicated staff of the Faculty of Computing for their consistent assistance and motivation throughout my academic journey. Their commitment to both teaching and research has been a true source of inspiration. Furthermore, I would like to thank our lecturer in charge, Dr. Jayantha Amararachchi and the CDAP team for their assistance in ensuring the project's success.

I also wish to express my appreciation to my fellow team members for their invaluable support and collaborative efforts. Our insightful discussions and idea exchanges have significantly enriched my comprehension of the subject matter.

I am also appreciative of my teammates' assistance and cooperation. Our conversations and idea sharing have improved my comprehension of the topic. I want to express my gratitude for the assistance from Mr. Gayan Ranaweera, our external supervisor who specializes in electronic engineering at MAS Linea Aqua, for his kind support of this research study in SeamSense. Their assistance has been crucial to the success of our study.

Finaly, I would like to thank everyone who took part in this research and gave their time and wisdom. Their willingness to offer their insights and experiences has been helpful in assisting me in learning more about the subject.

ABSTRACT

Enhancing Apparel Industry Quality Control with SeamSense: Fusion Modeling for Worker-Centric Defect Analysis: Integrating Traditional and Time-Series Approaches

In the competitive and quality-driven apparel manufacturing industry, minimizing defects is essential for maintaining product standards and operational efficiency. At MAS Linea Aqua, a leading company in the sector, defect prediction poses significant challenges due to the complex nature of garment production and the variability in worker performance. This research introduces SeamSense, a pioneering fusion modeling approach that combines traditional statistical methods with advanced time-series analysis to enhance worker-centric defect analysis.

SeamSense aims to leverage historical and real-time data to predict defect rates accurately, facilitating proactive defect management and prevention. By integrating worker demographic information—such as experience, training, and work hours—into the prediction model, this approach assesses the influence of individual workers on defect rates. This worker-centric model not only improves prediction accuracy but also supports targeted interventions to boost worker performance and reduce defects.

The research methodology involves collecting comprehensive worker demographic data and incorporating it into the prediction model to evaluate its impact on defect rates. Advanced forecasting techniques, including hybrid ARIMA and traditional machine learning models, will be used to predict future defect rates based on historical data and time-series analysis. The prediction model will be fine-tuned using machine learning techniques to optimize performance, accounting for both traditional statistical metrics and worker-specific factors.

Key stages of the project include defining research objectives, gathering and integrating worker demographic data, developing the prediction model, and evaluating its performance using real-world data from MAS Linea Aqua's production floor. By addressing the gap in worker-centric defect analysis and utilizing advanced modeling techniques, this research aims to provide actionable insights for defect management and process optimization in the apparel industry.

Keywords: Fusion modeling, Worker performance analysis, Defect prediction, Time-series forecasting, Apparel manufacturing, Production quality control, Seam defect management, Machine learning optimization, Worker demographic integration, Predictive analytics.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Table 1 - List of Abbreviations

Abbreviation	**Description**
TSA	Time Series Analysis
ARIMA	Auto Regressive Integrated Moving Average
ML	Machine Learning
SVM	Support Vector Machine
RF	Random Forest
TRM	Traditional Model
LR	Linear Regression
GB	Gradient Boosting
WBS	Work Breakdown Structure

1. INTRODUCTION

MAS [1] Holdings, founded in 1987 by Mahesh, Sharad, and Ajay Amalean, is a design-to- delivery solution provider in apparel and textile manufacturing. Initially focused on intimate apparel, the company later diversified into sportswear, performance wear, swimwear, brands, wearable technology, FemTech, start-ups, and industrial parks. Linea Aqua, a joint venture between Speedo International (UK), Brandot International (US), and MAS Holdings, aims to be the most compelling swimwear supplier globally and has become a key player in the swimwear industry.

In today's dynamic and competitive manufacturing landscape, the ability to accurately predict defect rates is paramount for ensuring product quality, optimizing production processes, and maintaining a competitive edge in the market. [2] Defect rate prediction plays a crucial role in enabling manufacturers to proactively identify potential issues, implement preventive measures, and minimize the impact of defects on product quality and customer satisfaction. Traditional approaches to defect rate prediction often rely on historical data analysis and statistical modeling techniques, which may lack the ability to capture the complex and dynamic nature of manufacturing processes.

To address these challenges and enhance defect rate prediction capabilities, this research proposal introduces "SeamSense," a comprehensive prediction system designed to integrate time series analysis, worker demographics, and stacked models. SeamSense aims to leverage the strengths of each component to develop more accurate and robust defect rate predictions tailored to the unique requirements of manufacturing environments. By combining advanced machine learning algorithms, fusion modeling techniques, and real-time data analysis, SeamSense seeks to revolutionize defect analysis methodologies and empower manufacturers with actionable insights for quality control and decision-making.

The significance of this research lies in its potential to bridge the gap between traditional defect rate prediction methods and the evolving needs of modern manufacturing operations. By harnessing the power of time series analysis, SeamSense

can capture temporal trends and patterns in defect rates, allowing manufacturers to anticipate fluctuations and proactively address potential issues. The integration of worker demographics into the prediction model enables SeamSense to account for the human element in manufacturing processes, providing a more holistic understanding of the factors influencing defect rates.

Through this research proposal, we aim to advance the field of defect analysis and quality control by introducing a novel approach that combines state-of-the-art machine learning techniques with domain-specific knowledge and real-time data analytics. The proposed project plan outlines the key objectives, methodologies, and deliverables, demonstrating our commitment to driving impactful advancements in predictive analytics and machine learning within the manufacturing sector. Together, we envision SeamSense as a game-changing solution that empowers manufacturers to achieve higher levels of quality, efficiency, and competitiveness in today's fast-paced manufacturing environment.

1.1. Background & Literature Survey

In today's manufacturing landscape, ensuring product quality, and minimizing defects are paramount objectives for companies striving to maintain competitiveness and meet customer expectations. Defect rate prediction plays a pivotal role in achieving these goals, providing insights into potential production issues, and facilitating proactive interventions. Defect rate prediction has relied on historical data analysis and statistical modeling techniques, which, although valuable, may not capture the dynamic nature of manufacturing processes and the multifaceted factors influencing defect occurrence.

The advent of advanced technologies, including ML, time-series analysis, and predictive analytics, presents new opportunities to enhance defect rate prediction methodologies. These technologies offer the potential to develop more sophisticated and adaptive prediction models capable of incorporating diverse data sources and capturing complex patterns in defect occurrence. Moreover, there is a growing recognition of the importance of considering human factors, such as worker demographics, in defect analysis. Factors such as experience, training, and work hours can significantly influence defect rates, yet they are often overlooked in traditional prediction models.

Against this backdrop, the proposed research seeks to address the limitations of existing defect rate prediction methods by integrating advanced technologies and worker-centric approaches. By leveraging time series analysis, machine learning algorithms, and insights from worker demographics, the research aims to develop a fusion modeling framework tailored for defect analysis in manufacturing. This framework will not only enhance the accuracy and reliability of defect rate predictions but also provide valuable insights into the underlying factors driving defects.

The significance of this research extends beyond its immediate application in manufacturing. The development of robust defect prediction models has implications for various industries, including automotive, aerospace, and electronics, where quality control and predictive maintenance are critical. By advancing defect rate prediction methodologies and integrating human-centric perspectives, the proposed research aims

to contribute to the broader discourse on predictive analytics and proactive quality management. The goal is to empower manufacturing organizations with the tools and insights needed to optimize production processes, minimize defects, and deliver high-quality products to consumers.

The textile industry is evolving with Industry 4.0, but [2] limited data sharing hinders advanced technologies like ML. Existing research lacks comprehensive data sharing for fault simulation and forecasting. [2] This study proposes a method using ML for fault prediction, showing promising accuracy with the random forest algorithm. It contributes to improving production efficiency and quality in the textile industry. This literature review sets the stage for exploring similar challenges and solutions in our component, "Fusion Modeling for Worker-Centric Defect Analysis: Integrating Traditional and Time-Series Approaches in Apparel Manufacturing." [2]

The introduction of advanced technologies, including machine learning (ML), time-series analysis, and predictive analytics, has opened new avenues for enhancing defect rate prediction methodologies. These technologies enable the development of more sophisticated and adaptive prediction models capable of incorporating diverse data sources and capturing complex patterns in defect occurrence. Zhang (2003) [3] highlighted the potential of hybrid models, combining ARIMA with artificial neural networks (ANNs), to improve forecasting accuracy by addressing both linear and nonlinear data relationships. This hybrid approach is particularly relevant for manufacturing, where both types of data patterns are present. [3]

Time series forecasting is vital for decision-making across domains. ARIMA, a traditional linear model, is widely used for its simplicity and effectiveness. Developed by Box and Jenkins, it excels in capturing linear trends and patterns in data, making it a cornerstone in time series analysis. [3] In contrast, Artificial Neural Networks (ANNs) offer powerful capabilities for capturing complex nonlinear relationships in data. Their flexibility makes them attractive for domains with dynamic data, where traditional linear models may fall short. Recent studies have explored hybrid approaches, combining ARIMA and ANN methodologies to leverage the strengths of both. The

proposed hybrid ARIMA-ANN model aims to enhance forecasting accuracy by integrating linear and nonlinear modeling techniques. Experimental results with real-world datasets demonstrate the effectiveness of the hybrid model, outperforming individual models. This suggests that hybrid approaches could offer a more robust and accurate forecasting solution for decision-making processes. [3]

Wu et al. (2023) [4] introduced the TWC-EL model, which integrates three-way clustering and ensemble learning to enhance prediction accuracy in complex scenarios. [4] This model's ability to categorize data into core and fringe regions, followed by integration into an ensemble prediction framework, has shown significant advancements in dealing with intricate data structures. Such innovative approaches are essential for improving prediction accuracy in manufacturing environments where multiple interacting variables impact defect rates. [4]

In traditional defect prediction models, the influence of human factors, such as worker demographics, has often been overlooked. However, these factors, including experience, training, and work hours, can significantly influence defect rates. Juran and Schruben (2004) [5] emphasized the importance of incorporating worker personality traits and demographic information into predictive models to improve system performance. This worker-centric approach aligns with the goals of the proposed research, which seeks to integrate these human factors into defect prediction models to provide more accurate and actionable insights. [5]

The impact of integrating worker data into predictive models was further explored by Seçkin et al. (2019)**,** [5] who applied machine learning techniques to simulate and forecast production faults in the textile industry. Their findings demonstrated that incorporating worker-specific data can enhance the accuracy of fault prediction, which is critical for improving production efficiency and reducing defects in manufacturing. The apparel industry, despite being a significant part of global manufacturing, faces challenges related to data sharing and the adoption of advanced technologies. Seçkin et al. (2019) [5] highlighted these challenges in their study on fault simulation and forecasting in the glove textile industry, showing that machine learning models,

particularly those using random forest algorithms, can significantly improve production efficiency and quality by accurately predicting faults. This research is particularly relevant to the proposed study, which focuses on integrating traditional and advanced modeling techniques for defect analysis in apparel manufacturing. [5]

Moreover, Liu et al. (2023) [6] proposed a fusion model combining ARIMA, LSTM, and Prophet to enhance time-series forecasting accuracy. The P-gLSTNet model demonstrated superior performance compared to traditional methods, showcasing the potential of fusion models in handling complex time-series data. These findings are directly applicable to manufacturing, where accurate prediction models are essential for maintaining high-quality production standards. [6]

The implications of developing robust defect prediction models extend beyond the immediate application in apparel manufacturing. Industries such as automotive, aerospace, and electronics, where quality control and predictive maintenance are critical, can benefit from advancements in defect rate prediction methodologies. By incorporating human-centric perspectives and leveraging advanced modeling techniques, this research aims to contribute to the broader discourse on predictive analytics and proactive quality management. The fusion of traditional statistical models with machine learning approaches offers a comprehensive framework for improving defect prediction across various manufacturing sectors, ultimately leading to enhanced product quality and operational efficiency. [7]

The study on machine learning strategies for time series forecasting provides valuable insights directly relevant to our component on defect prediction in apparel manufacturing. By exploring various machine learning techniques for forecasting future behavior based on historical data, it addresses the need for robust and efficient predictive models—an essential requirement in manufacturing defect prediction. The chapter's focus on formalizing forecasting problems as supervised learning tasks and discussing the role of forecasting strategies aligns with our research objectives of developing accurate predictive models for defect rates in apparel manufacturing. Therefore, this literature review serves as a foundational reference for our research,

guiding us in selecting appropriate machine learning techniques for defect prediction. [8]

The study explores the impact of worker personality and demographic information on system performance prediction, offering insights applicable to our defect prediction research in apparel manufacturing. It highlights the importance of accurately representing individual worker behavior in predictive modeling, aligning with our objective of integrating worker demographics into defect prediction models [9]. The practical implications for managerial processes provide valuable guidance for our research, informing our approach to incorporating demographic information into predictive modeling. [9]

The study focuses on minimizing defects in the sewing department of apparel manufacturing, a critical aspect in our research on defect prediction. It underscores the challenges faced due to human error and the need for continuous quality improvement. [9] The emphasis on statistical analysis and root cause analysis aligns with our objective of developing predictive models to identify and prevent defects. The discussion on the consequences of defects, such as wasted production costs, highlights the significance of defect prediction and prevention. Keywords like "Quality control," "Data analysis," and "Defects" resonate with our research goals, emphasizing the relevance of this study to our component. [10]

1.2. Research Gap

Identifying research gaps is crucial for any research endeavor as it helps pinpoint areas of unexplored knowledge within a given field. These gaps serve as focal points, guiding researchers in defining the scope of their study and uncovering new avenues for inquiry. In our proposed system, Fusion Modeling for Worker-Centric Defect Analysis: Integrating Traditional and Time-Series Approaches, several distinctive features have been identified. These include novel compression strategies for minimizing file size and the innovative integration of traditional and time-series approaches for worker-centric defect analysis. To discern these gaps and explore new research directions, we conducted a comparative analysis between our proposed system and a selection of relevant research papers. This analysis serves as a roadmap, guiding our efforts to innovate and advance the field of worker-centric defect analysis.

In contrast to the proposed system, the selected research papers were scrutinized to pinpoint areas where existing knowledge falls short, thereby identifying potential research gaps. This comparative analysis shed light on untapped opportunities and unaddressed challenges within the domain, prompting further exploration and investigation. By juxtaposing the proposed system with existing research, we were able to delineate avenues for innovation and expansion, thus enriching the research landscape and propelling the field forward. Through this process, we aim to contribute to the advancement of knowledge in our chosen subject area while addressing pertinent research gaps.

Although textile production is heavily automation-based, it is viewed as a virgin area about Industry 4.0. [3] Efficiency gains are anticipated when the innovations are applied to the textile industry. Studies on data mining and machine learning in the textile industry reveal that businesses lack the data sharing necessary to share information about their production processes due to confidentiality and financial considerations. This paper presents a machine learning method for producing regression from time series data by simulating a production process. For the annual production plan, a simulation has been created, and the related errors have been identified using production data and information from the textile glove industry. In

the context of supervised learning, a data collection has been used to test different machine learning techniques and compare the learning. Random parameters in the simulation have been used to create the errors that occur in the production process. A variety of machine learning methods have been taught using time series data sets to validate the idea that the errors could be predicted. It was possible to predict the variable indicating the quantity of defective products with great success. The random forest algorithm has shown the highest level of performance.

when it comes to forecasting the defective product parameter. Because these error values have produced great accuracy even in a simulation with evenly distributed random parameters, real-world applications can also produce extremely accurate forecasts. In the domain of defect prediction within manufacturing, existing literature predominantly focuses on either time series analysis or worker demographics separately, neglecting the potential synergies that can arise from integrating both aspects. This presents a significant research gap that warrants exploration, particularly in the context of apparel manufacturing, such as MAS Linea Aqua. Through the integration of time series analysis techniques with worker demographic data, researchers can unlock novel insights into defect prediction dynamics and enhance the accuracy of predictive models.

A key research gap exists in the absence of studies that comprehensively TSA and worker demographics for defect prediction in apparel manufacturing. While some papers concentrate on time series forecasting using historical data, they often overlook the influence of worker characteristics on defect rates. Studies examining the impact of worker demographics on system performance prediction may fail to fully capitalize on time series trends in defect prediction. Bridging this gap necessitates a holistic approach that incorporates both temporal patterns and worker-related factors in defect prediction models.

Existing research primarily addresses fault simulation and prediction in general manufacturing settings, with limited attention to the unique challenges of apparel manufacturing. The production processes at MAS Linea Aqua are characterized by

intricate garment construction and diverse materials, necessitating tailored defect prediction models. There is a pressing need for research specifically addressing the complexities of defect prediction in apparel manufacturing, drawing insights from both time series analysis and worker demographics.

While some studies propose hybrid forecasting techniques for defect prediction, their application in apparel manufacturing remains largely unexplored. [4] Hybrid models that amalgamate traditional forecasting methods with machine learning algorithms offer potential enhancements in prediction accuracy.

However, adapting these techniques to the dynamic nature of apparel production processes necessitates further investigation. [6] The optimization of stacking approaches for defect prediction in apparel manufacturing presents another research gap. While stacking techniques exhibit promise in amalgamating the strengths of multiple models, their application to defect prediction models in apparel manufacturing settings is still limited. [6] Optimizing stacking approaches to leverage worker demographics and time series data could yield more accurate defect prediction models tailored to the production environment at MAS Linea Aqua.

Table 2 - Overall Research Gap

Research Gap Feature	**Research 1** [3]	**Research 2** [5]	**Research 3** [4]	**Research 4** [6]	**Research 5** [10]	**Proposed Research Solution**
Predictive Forecasting with Real-Time Adaptation	✓	✗	✓	✗	✗	✓
Worker Demographic Predictive Analytics Integration	✗	✗	✗	✗	✓	✓
Innovative Ensemble Methods in apparel industry	✗	✓	✗	✓	✗	✓
Synergistic Model Stacking	✗	✗	✗	✓	✗	✓
Dynamic Data Fusion for Enhanced Predictions / adaptive iterative refinement	✗	✗	✓	✗	✗	✓
Comprehensive system combining multiple advanced models	✗	✓	✗	✗	✗	✓
Long term defect rate forecasting capability	✗	✗	✗	✗	✗	✓

2. RESEARCH PROBLEM

Research Problem: My component's research problem revolves around the need to enhance defect rate prediction in manufacturing processes. Specifically, the challenge lies in developing accurate and reliable prediction models that can effectively integrate time-series analysis and worker demographic information. Traditional approaches often overlook the complex interplay between various factors influencing defect rates, including production parameters and human factors like worker demographics. The research problem centers on addressing these gaps by leveraging advanced modeling techniques to improve defect rate predictions and optimize manufacturing operations.

The integration of advanced technologies such as machine learning (ML), time-series analysis, and predictive analytics presents new opportunities to enhance defect prediction methodologies. However, existing research predominantly focuses on either time-series analysis or worker demographics in isolation, neglecting the potential synergies that could arise from integrating these aspects. This fragmented approach fails to leverage the full potential of predictive models in providing accurate and actionable insights into defect dynamics.

The research problem addressed by my component revolves around the need for accurate and reliable defect rate prediction in manufacturing processes. In today's industrial landscape, minimizing defects is paramount for ensuring product quality, optimizing production efficiency, and reducing costs. Traditional defect prediction methods often lack the precision and adaptability required to meet the dynamic demands of modern manufacturing environments.

One of the key challenges is the complex interplay between various factors that influence defect rates, including production parameters, environmental conditions, and, importantly, human factors such as worker demographics. While time-series analysis has shown promise in forecasting defect rates based on historical data, integrating worker demographic information into prediction models remains largely unexplored territory. Understanding how worker characteristics such as experience, training, and

work hours impact defect rates is crucial for developing more accurate and robust prediction models.

Existing machine learning approaches often fail to leverage the full potential of diverse data sources and advanced modeling techniques. Traditional models may overlook subtle patterns in defect trends or struggle to adapt to changing production conditions. There is a pressing need for innovative fusion models that can effectively combine time-series analysis, worker demographics, and stacked learning techniques to enhance defect rate predictions.

Addressing this research problem requires a multifaceted approach that encompasses data collection, modeling, and iterative refinement. By leveraging the strengths of time-series analysis and machine learning, our research aims to develop a fusion modeling framework that can seamlessly integrate disparate data sources and adapt to evolving manufacturing dynamics. Through this endeavor, we seek to advance the state-of-the-art in defect prediction methodologies and contribute to the optimization of manufacturing processes.

3. OBJECTIVES

3.1. Main Objective

The main objective of the research component is to develop a comprehensive prediction system called SeamSense, which integrates various methodologies to improve defect rate prediction accuracy in manufacturing processes. This system aims to address the challenges associated with traditional defect rate prediction methods by leveraging the strengths of time series analysis, worker demographics, and stacked models. SeamSense seeks to provide manufacturing industries with a sophisticated tool that can effectively forecast defect rates, thereby enabling proactive quality control measures and optimization of production processes. By combining diverse techniques and data sources, SeamSense aims to enhance prediction accuracy and provide valuable insights into the factors influencing defect rates in manufacturing environments. The main objective is to develop a versatile and reliable prediction system that empowers manufacturers to maintain high product quality standards and improve operational efficiency

3.2. Sub Objectives

- Collect demographic information of workers, including factors such as experience, training, and work hours, to incorporate into the prediction model:

Specific - Define the specific demographic variables to be collected, such as years of experience, types of training received, and average weekly work hours, ensuring clarity and consistency in data collection efforts.

Measurable - Develop clear metrics for each demographic variable to be collected, allowing for quantitative assessment and comparison across different worker profiles.

Achievable - Implement user-friendly data collection methods, such as online surveys or digital forms, to facilitate the efficient gathering of demographic information from workers in various roles and departments.

Relevant - Ensure that the collected demographic data directly aligns with known factors influencing the prediction of outcomes in the targeted domain, such as defect rates in manufacturing.

Time-bound - Establish a timeline for completing the data collection process, setting specific deadlines for gathering each demographic variable to ensure timely incorporation into the prediction model.

- Utilize forecasting techniques to predict future defect rates based on historical data and time-series analysis results:

Specific - Select appropriate forecasting techniques, such as ARIMA (Autoregressive Integrated Moving Average) or Exponential Smoothing, to analyze historical defect rate data and derive future predictions.

Measurable - Evaluate the accuracy of the forecasting techniques by comparing predicted defect rates with actual observed rates over time, using quantitative metrics like Mean Absolute Error (MAE) or Root Mean Square Error (RMSE).

Achievable - Implement forecasting algorithms within the prediction system, leveraging available software libraries or custom code, ensuring compatibility with the historical data and analysis results.

Relevant - The use of forecasting techniques directly aligns with the objective of predicting future defect rates, providing valuable insights for proactive quality management and decision-making in manufacturing.

Time-bound - Establish a timeline for completing the forecasting tasks, including deadlines for data preprocessing, model training, validation, and deployment, ensuring that predictions are generated in a timely manner to support operational planning and management.

- Integrate worker demographic features into the prediction model to assess their impact on defect rates and enhance prediction accuracy:

Specific - Identify specific worker demographic features to be integrated into the prediction model, such as age, education level, tenure, and job role, ensuring clarity and specificity in defining the variables.

Measurable - Quantify the impact of worker demographic features on defect rates by analyzing the model's performance metrics before and after the integration of demographic data, such as changes in prediction accuracy or reduction in prediction errors.

Achievable - Develop a methodology for integrating demographic features into the prediction model, leveraging appropriate statistical techniques or machine learning algorithms, and ensuring compatibility with the existing model architecture.

Relevant - The integration of worker demographic features directly addresses the research objective of understanding the factors influencing defect rates in manufacturing, providing valuable insights for process optimization and quality control.

Time-bound - Establish a timeline for completing the integration process, including tasks such as data preprocessing, feature engineering, model training, and evaluation, with clear deadlines to ensure timely completion within the project timeline.

- Fine-tune the stacked model, leveraging the strengths of time series analysis, traditional machine learning approaches, and pattern recognition techniques, to optimize predictive performance:

Specific - Identify specific parameters and hyperparameters within the stacked model architecture that can be fine-tuned, such as learning rates, regularization strengths, and ensemble weights, to optimize predictive performance.

Measurable - Quantify the improvement in predictive performance through objective evaluation metrics, such as accuracy, precision, recall, or F1-score, before and after fine-tuning the stacked model.

Achievable - Implement fine-tuning algorithms and techniques, such as grid search or random search, to systematically explore the hyperparameter space and identify optimal configurations for the stacked model.

Relevant - Fine-tuning the stacked model aligns with the research objective of optimizing predictive performance by leveraging the strengths of different modeling approaches, enhancing the model's ability to accurately forecast defect rates in manufacturing.

Time-bound - Establish a timeline for the fine-tuning process, including deadlines for conducting hyperparameter optimization experiments, evaluating model performance, and integrating the optimized model into the prediction system, ensuring timely completion within project milestones.

4. METHODOLOGY

4.1. Requirement Gathering and Analysis

Requirement gathering is a critical phase in the development of the "Fusion Modeling for Worker-Centric Defect Analysis" system. This phase involves collecting, analyzing, and documenting the necessary requirements to ensure that the system meets the needs of its stakeholders and functions as intended. The process is divided into several key steps:

- **Collecting Related Research Papers:** Gather relevant academic and industry research papers to understand existing approaches and identify gaps.
- **Conducting a Feasibility Study:** Assess the technical, economic, and operational feasibility of the proposed system.
- **Conducting a Background and Literature Assessment:** Review the collected literature to gain insights into current methodologies and their limitations.
- **Reading and Evaluating Research Papers:** Analyze the selected research papers to extract key findings and relevant methodologies.
- **Gathering Data from Users:** Engage with users and stakeholders to understand their needs and expectations and gather real-world data.
- **Evaluating User Perspectives:** Analyze user feedback to ensure the system design aligns with their requirements and expectations.
- **Identifying Suitable Components:** Based on research and user input, determine the most appropriate system components and finalize the project scope.

4.1.1. Functional Requirements

- **Data Collection and Integration:**

The system must collect and integrate diverse data sources, including worker demographics, historical defect rates, and production variables, for defect rate prediction. It should retrieve data from databases and preprocess it for quality and compatibility. This ensures comprehensive and reliable data for accurate predictions.

- **Data Collection Capability:**

SeamSense must be able to collect a variety of data types, including worker demographics (age, experience, training history), machine data (operational hours, maintenance schedules), and production data (output rates, defect rates).

- **Time Series Analysis:**

SeamSense must perform statistical time series analysis to understand defect trends over time and forecast future defect occurrences.

- **Demographic Data Integration:**

The system should have the capability to integrate worker demographic data with production and defect data to assess potential correlations and influences on product quality.

- **Predictive Modeling:**

SeamSense should use stacked models to make predictions about future defect rates, incorporating both time series data and demographic information.

- **Feedback Mechanism:**

The system must iteratively refine its models based on feedback from the production data and potentially other sources such as quality control feedback.

- **Reporting and Alerts:**

SeamSense should generate reports and alerts for predicted high defect rates, allowing preemptive actions to mitigate quality issues.

- **Historical Data Analysis:**

The system must allow users to analyze historical data to identify long-term trends and assess the effectiveness of past interventions.

- **Model Stacking and Ensemble Learning:**

The system should implement model stacking and ensemble learning techniques to improve the robustness and accuracy of defect predictions. This involves combining the strengths of multiple models.

- **Scalability:**

The system should be scalable to handle large volumes of data and support multiple production lines or facilities.

4.1.2. Non-Functional Requirements

- **Performance:**

SeamSense must process data and provide predictions in a timely manner, with high throughput and low latency.

- **Accuracy:**

The predictions made by the system must be highly accurate and reliable.

- **Scalability:**

The system should be scalable, able to handle increasing amounts of data as the production line expands or as new data sources are integrated.

- **Usability:**

SeamSense should have a user-friendly interface for inputting data and interpreting predictions and reports.

- **Security:**

Worker demographic data and production data should be securely stored and processed, with access controls in place to prevent unauthorized access.

- **Maintainability:**

The system must be easy to maintain and update, with clear documentation and support for troubleshooting and enhancements.

- **Compatibility:**

SeamSense should be compatible with existing systems and technologies used in the production environment.

- **Interoperability:**

The system must be able to integrate with existing manufacturing systems, databases, and third-party software. It should support standard data formats and protocols to facilitate seamless data exchange.

4.2. Feasibility Study

A feasibility study is a critical assessment that evaluates the viability of a proposed project by analyzing various aspects such as technical, operational, economic, and legal factors. This study will assess the feasibility of developing a "Fusion Modeling for Worker-Centric Defect Analysis: Integrating Traditional and Time-Series Approaches" system in the context of apparel manufacturing.

Technical Feasibility:

The technical feasibility focused on evaluating the availability of required technologies and resources for system development and integration. It confirmed that the necessary modeling techniques, such as machine learning (ML), time-series analysis, and predictive analytics, are mature and widely available. The study also ensured that the integration frameworks are compatible with existing manufacturing systems, allowing for seamless implementation of the proposed approach.

Economic Feasibility:

Economic feasibility was assessed by estimating the costs associated with hardware, software, data collection, and personnel salaries. A cost-benefit analysis was conducted to weigh the potential benefits, such as improved defect prediction accuracy and operational efficiency, against the projected expenses. The analysis indicated a favorable return on investment (ROI), suggesting that the long-term benefits outweigh the initial and ongoing costs.

Operational Feasibility:

Operational feasibility analyzed the impact of the proposed system on workflow efficiency, user acceptance, and organizational readiness for change. Interviews and surveys with stakeholders, including management and workers, were conducted to gauge their willingness to adopt the new system and identify potential barriers to implementation, such as resistance to change or the need for additional training. The study found that with proper training and change management strategies, the

system could be successfully integrated into existing workflows, leading to enhanced productivity and quality control.

4.3. System Design

4.3.1. High-level Overall System Architecture Diagram

The diagram illustrates a real-time defect detection system for apparel manufacturing, integrating advanced technologies like high-definition video analysis, fog-cloud computing, machine learning, and fusion modeling. High-definition cameras capture seam footage, which is processed using image analysis techniques to detect defects. Fog-cloud computing efficiently manages video data processing and storage, while a machine learning model analyzes the data in real-time to predict defect types. The system also incorporates worker data and time-series analysis to enhance prediction accuracy. A feedback loop allows supervisors to monitor and adjust production processes, ensuring continuous quality improvement. This integrated system is scalable and applicable across various manufacturing sectors.

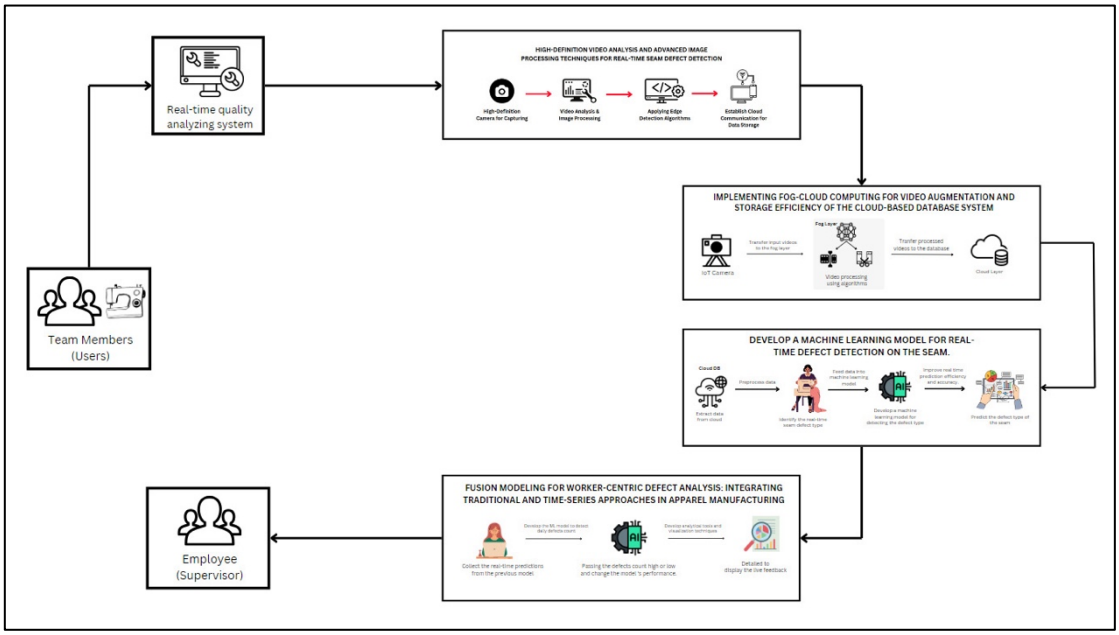


Figure 1 - Overall System Architecture

4.3.2. Component Diagram

The architecture for the "Fusion Modeling for Worker-Centric Defect Analysis" system integrates traditional machine learning models with advanced time-series forecasting to enhance defect prediction in manufacturing. It collects real-time predictions and worker demographic data to improve accuracy. A time-series model forecasts trends based on historical data, while a traditional ML model analyzes how worker factors influence defects. These models are combined using an ensemble method to produce robust

predictions. User feedback is used to refine the system, ensuring continuous improvement. This architecture effectively combines historical and real-time data to optimize defect management in manufacturing.

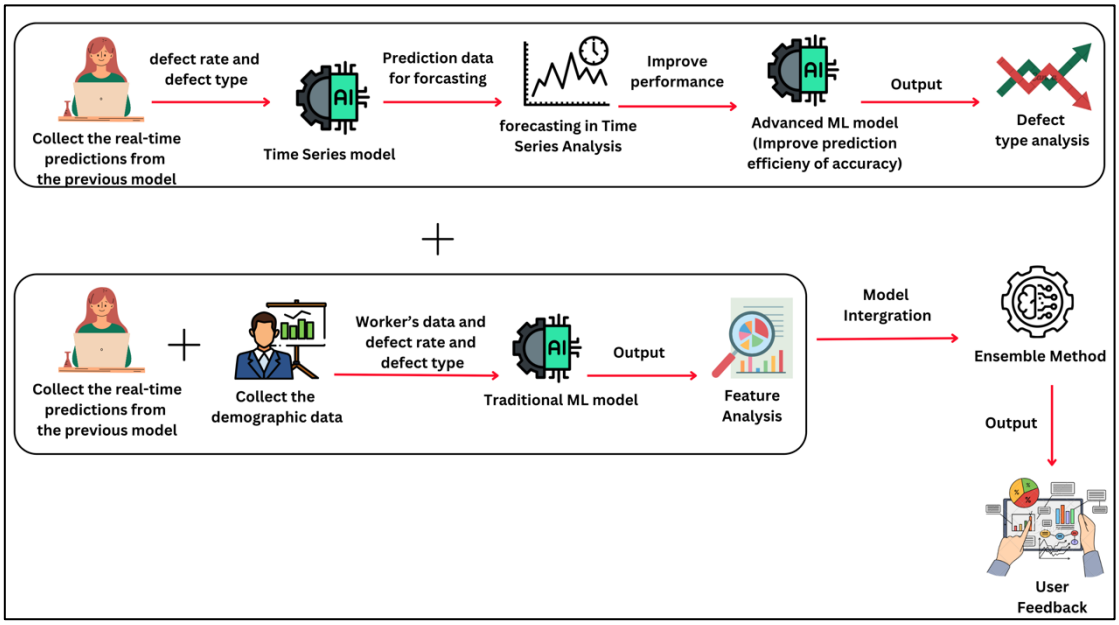


Figure 2 - Component Diagram

4.4. Dataset Description and Data Collection

The datasets utilized in this research component are designed to analyze and predict worker-related defects in an industrial context. Two primary datasets are employed: the **Demographic Data** and the **Defect and Production Data**. The **Demographic Data** includes critical information about the workers, such as their unique identifiers, joining dates, ages, genders, skill levels, and shifts. This data is instrumental in understanding how worker characteristics may influence defect occurrences. The **Defect and Production Data** records the defects encountered, and the production volumes achieved by each worker on specific dates. Key defect types such as 'Run_Off', 'Open_Seam', 'SPI_Errors', and 'High_Low' are tracked alongside the worker's production output and the shift during which these defects occurred.

The datasets are linked through the Worker_ID field, allowing for a comprehensive analysis that integrates demographic characteristics with defect occurrences and production volumes. The time-series nature of the data, with the **Date** field, enables trend analysis over time, making it possible to forecast future defect occurrences. However, challenges such as missing data, data imbalance, and the complexity of time-series forecasting must be carefully addressed to ensure the accuracy and reliability of the predictive models developed in this research. Overall, these datasets provide a robust foundation for understanding the factors contributing to production defects and for developing predictive models to improve industrial quality control processes.

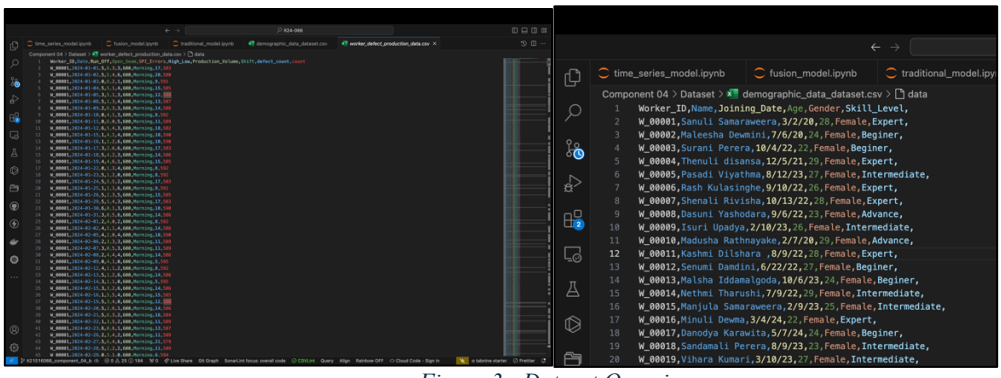


Figure 3 - Dataset Overview

4.5. Understanding the Key Pillars

In the context of the "Fusion Modeling for Worker-Centric Defect Analysis" system, the understanding and key pillars provide the foundation for its development and implementation. The system is designed to enhance defect prediction accuracy in manufacturing by integrating traditional machine learning models with advanced time-series forecasting techniques. This understanding forms the basis for the system's objective to not only predict defects but also to incorporate worker-centric data, recognizing the significant role human factors play in manufacturing quality.

The key pillars supporting this system include data integration, predictive modeling, ensemble methods, user-centric design, and continuous improvement. Data integration ensures that all relevant data, including real-time production metrics and worker demographics, are seamlessly brought together to provide a comprehensive view of the factors influencing defects. Predictive modeling involves the use of advanced machine learning and time-series analysis to identify patterns and trends, enabling more accurate and timely defect predictions. The system's robustness is further enhanced by ensemble methods, which combine the outputs of multiple models to deliver more reliable predictions.

A User-centric design ensures that the insights generated are accessible and actionable, with a focus on ease of use for operators and supervisors. Finally, continuous improvement is a core pillar, with the system designed to adapt and evolve based on real-time feedback and changing production conditions. Together, these key pillars provide a strong framework that supports the system's goal of optimizing quality control processes and improving manufacturing outcomes.

4.6. Tools and Libraries

The table presented below offers a comprehensive account of the tools and technologies employed in the development of the application as well as the program.

Table 3 – Tools and Technologies

Category	Tools and Technologies	Description
Programming Languages	Python	Used for developing machine learning models, data processing, and building the Flask web app.
Machine Learning Libraries	Scikit-learn	Implemented traditional machine learning models like Gradient Boosting Regressor and preprocessors.
	Statsmodels	Developed ARIMA models for time-series forecasting.
	joblib	Used for saving and loading machine learning models.
Data Handling and Processing	Pandas	Handled data manipulation, loading, cleaning, and transformation.
	NumPy	Supported numerical operations and handled large arrays efficiently.
Visualization	Matplotlib	Plotted results of models, including forecasted defect rates.
	Seaborn	Enhanced data visualizations and statistical plots.
Web Development	Flask	Developed a web application for user interaction and displaying model predictions.
	HTML/CSS	Structured and styled the web pages served by the Flask application.
Database	MongoDB	Stored worker and defect data, providing efficient retrieval and storage.
Deployment	Docker	Hosted and deployed the Flask web application.
Version Control	Git	Tracked changes, collaborated, and maintained the history of the project.
	GitHub	Hosted the project's code repository for version control and collaboration.
IDE (Integrated Development Environment)	Visual Studio Code	Used for writing and testing code, managing the project structure.

4.7. Methodology

The primary objective of this research is to predict defect occurrences in an industrial setting by leveraging time-series forecasting models, traditional machine learning techniques, and a novel fusion approach. The methodology is divided into several key stages, including data collection, preprocessing, model development, and deployment.

1. Data Collection

Data was collected from an industrial setup, focusing on worker defect logs and production data. The dataset included:

- **Worker Demographic Data**: Contains attributes like Worker ID, Name, Age, Gender, Skill Level, and Joining Date.
- **Defect and Production Data**: Includes records of various defect types (e.g., Run_Off, Open_Seam, SPI_Errors, High_Low) along with production volumes, timestamps, and associated worker IDs.

These datasets were integrated to form a comprehensive view of the factors influencing defect rates.

2. Data Preprocessing

Preprocessing involved several steps to ensure the datasets were suitable for analysis:

- **Date Parsing**: Converted date fields into datetime objects to facilitate time-series analysis.
- **Data Cleaning**: Handled missing values and removed duplicates to ensure data integrity.
- **Feature Engineering**: Categorical features such as Gender, Skill Level, and Shift were one-hot encoded. Numerical features like Age and Production Volume were standardized.
- **Time-Series Preparation**: Sorted data by date and transformed it into time-series format, setting dates as indices.

3. Model Development

Time-Series Forecasting with ARIMA

ARIMA (AutoRegressive Integrated Moving Average) models were employed to predict the occurrence of each defect type over time:

- **Model Training**: Individual ARIMA models were trained for each defect type using historical data.

- **Forecasting**: Each model was used to predict defect occurrences for a specified number of future time steps.

Traditional Machine Learning Models

Traditional machine learning models were developed to predict defects based on worker demographics and production data:

- **Model Selection**: MultiOutputRegressor with Gradient Boosting Regressor was selected for its ability to handle multiple output variables (defect types).
- **Cross-Validation**: Performed cross-validation to evaluate models and selected the best-performing model based on Mean Squared Error (MSE).

Fusion Model

To enhance predictive accuracy, a fusion model was developed by combining the outputs of the ARIMA time-series models with the predictions from the traditional machine learning model:

- **Model Integration**: Predictions from the ARIMA models were combined with the traditional model's predictions to create a comprehensive forecast.
- **Model Training**: The fusion model was trained using MultiOutputRegressor to optimize the combined predictions.

4. Model Evaluation

The models were evaluated using various metrics:

- **Mean Squared Error (MSE)**: Used to assess the accuracy of predictions for each defect type.
- **R^2 Score**: Evaluated the proportion of variance in the defect occurrences explained by the models.
- **RMSE**: Provided insights into the model's prediction error magnitude.

5. Deployment

The final models were deployed as a Flask web application, allowing users to input worker IDs and forecast steps to predict future defect occurrences:

- **Model Serialization**: The trained models were serialized using joblib for easy loading and inference in the web application.
- **User Interface**: A simple web UI was developed using Flask, where users can select worker IDs and specify the number of forecast steps.

- **Chart Visualization**: Predictions were visualized as charts using Matplotlib, enabling users to easily interpret the forecasted defect trends.

6. Database Integration

Data was stored and retrieved from a MongoDB database, providing efficient storage and access to worker and defect records:

- **Data Storage**: MongoDB was used to store the worker, defect, and forecast data.
- **Querying**: The Flask application queried the database to fetch relevant data for model inference.

4.8. Model Architecture

The model architecture for the defect prediction system is designed to integrate both traditional machine learning techniques and time-series analysis to forecast defect counts in a manufacturing environment. The process begins with data input, including historical defect data, worker demographics, and production metrics. This data undergoes preprocessing, where it is cleaned, features are engineered, and relevant attributes are encoded and scaled. The Linear Regression model, selected as the best-performing traditional machine learning model, is trained on this processed data to predict defect counts.

This research explores various traditional machine learning models for predicting defect rates in manufacturing, focusing on Random Forest, Gradient Boosting, Support Vector Regressor (SVR), Linear Regression, Ridge Regression, and Lasso Regression. Among these, Lasso Regression emerged as the best performer due to its effective feature selection and reduced risk of overfitting, making it suitable for scenarios with complex relationships between defect rates and features.

To enhance predictive accuracy, the stacking ensemble method is employed, combining the strengths of Lasso Regression with other models like Gradient Boosting and Random Forest. This ensemble approach creates a meta-model that integrates predictions from multiple models, leading to more robust and accurate predictions in a dynamic manufacturing environment.

In parallel, ARIMA models are utilized for time-series analysis, forecasting future defect trends based on historical patterns. The outputs from both the traditional model and the time-series models are then combined in a fusion layer, using a Gradient Boosting Regressor to enhance prediction accuracy. The final output consists of both predictions and visualizations of defect trends, which are then deployed via a Flask web application, allowing for real-time prediction and visualization. This integrated approach effectively combines static and temporal data, providing robust predictions and actionable insights for improving quality control processes.

4.8.1. Traditional Model

Figure 4 - Import Libraries for Time Series Model

os, NumPy, pandas: These libraries handle file paths, numerical operations, and data manipulation.

matplotlib, seaborn: Used for data visualization.

sklearn: A comprehensive library for machine learning tasks, including model creation, training, evaluation, and hyperparameter tuning.

scipy.stats: Provides statistical functions, used here for defining parameter distributions during hyperparameter tuning.

joblib: Used to save and load trained models.

Figure 5 - Save Model

os.makedirs(): Creates a directory to save the models if it doesn't already exist.

Figure 6 - Preprocessing the data

pd.read_csv(): Loads the demographic and defect datasets into Pandas DataFrames.

pd.to_datetime(): Converts the Joining_Date and Date columns to datetime objects for easier manipulation and analysis.

Figure 7 - Combining Datasets

pd.merge(): Merges the demographic data and defect data on the Worker_ID column to create a combined dataset.

Figure 8 - Dropping Unnecessary Columns

.drop(): Removes columns that are not needed for the analysis (Name, Joining_Date).

```
# Rename defect columns
defect_columns_mapping = {
    'Run_Off_D1': 'Run_Off',
    'Open_Seam_D2': 'Open_Seam',
    'SPI_Errors_D3': 'SPI_Errors',
    'High_Low_D4': 'High_Low'
}
combined_data.rename(columns=defect_columns_mapping, inplace=True)
```

Figure 9 - Rename Defect Columns

.rename(): Simplifies the column names related to defect types for easier access and readability.

```
# Define categorical features for one-hot encoding
categorical_features = ['Gender', 'Skill_Level', 'Shift']
numerical_features = ['Age', 'Production_Volume']

# Time-Series Feature Engineering
combined_data['DayOfWeek'] = combined_data['Date'].dt.dayofweek
combined_data['WeekOfYear'] = combined_data['Date'].dt.isocalendar().week
combined_data['Month'] = combined_data['Date'].dt.month
combined_data['Quarter'] = combined_data['Date'].dt.quarter

numerical_features.extend(['DayOfWeek', 'WeekOfYear', 'Month', 'Quarter'])
```

Figure 10 - Feature Engineering

categorical_features, numerical_features: Lists of features to be used in the model.

Time-Series Feature Engineering: Creates new features based on the Date column to capture temporal patterns (DayOfWeek, WeekOfYear, Month, Quarter).

```
# Ensure that all columns are present
missing_columns = [col for col in categorical_features + numerical_features if col not in combined_data.columns]
if missing_columns:
    raise ValueError(f"The following columns are missing in the combined dataset: {missing_columns}")
```

Figure 11 - Checking for Missing Values

missing_columns: Ensures that all necessary columns are present in the dataset, raising an error if any are missing.

```
# Define preprocessing steps
preprocessor = ColumnTransformer(
    transformers=[
        ('onehot', OneHotEncoder(), categorical_features),
        ('num', StandardScaler(), numerical_features)
    ]
)
```

Figure 12 -Data Preprocessing

ColumnTransformer: Applies different preprocessing steps to different subsets of features (OneHotEncoder for categorical features and StandardScaler for numerical features).

```
# Separate features and target variables
X = combined_data.drop(columns=['Run_Off', 'Open_Seam', 'SPI_Errors', 'High_Low', 'defect_count', 'count', 'Worker_ID', 'Date'])
y = combined_data[['Run_Off', 'Open_Seam', 'SPI_Errors', 'High_Low']]
```

Figure 13 - Splitting Data into Features and Target Variables

X: Feature variables.

y: Target variables (defect types).

```
# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figure 14- Splitting Data into Training and Testing Sets

train_test_split(): Splits the data into training and testing sets (80% training, 20% testing).

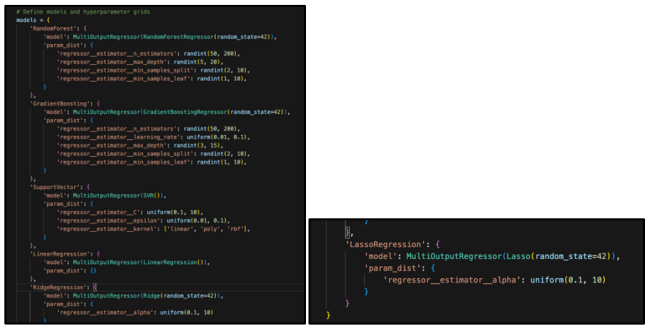


Figure 15 - Defining Models and Hyperparameter

models: A dictionary where each key represents a different model, and the corresponding value contains the model and its hyperparameter grid for tuning.

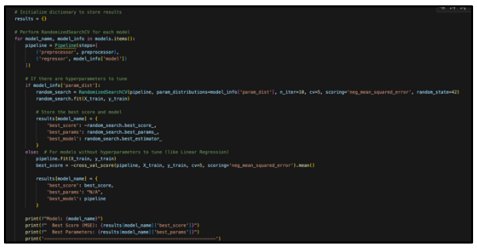


Figure 16 - Model Training

results: Stores the best model and its performance metrics.

RandomizedSearchCV: Performs hyperparameter tuning for models that have parameters to optimize.

Pipeline: Combines the preprocessor and model into a single object for streamlined training and evaluation.

Figure 17 - Saving Best Model

best_model_name: Identifies the model with the best performance.

joblib.dump(): Saves the best model to disk for future use.

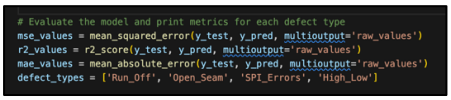


Figure 18 - Model Evaluation and Test set

mean_squared_error, r2_score, mean_absolute_error: Calculates evaluation metrics for the test set predictions.

Figure 19 - Visualization of Evaluation Metrics

plt.bar(): Creates bar plots to visualize the evaluation metrics for each defect type.

Figure 20 - Residual Analysis

Residuals: The difference between the actual and predicted values, used to assess the model's accuracy.

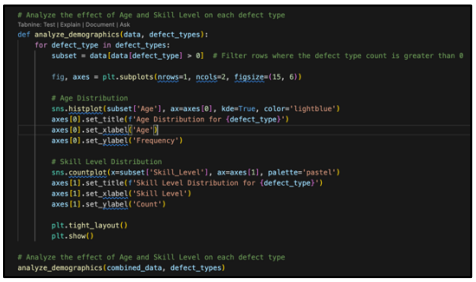


Figure 21 - Demographic Analysis

analyze_demographics(): Analyzes the impact of demographic factors (Age, Skill_Level) on defect types using histograms and count plots.

Figure 22 - Binary Classification Metrics

to_binary_predictions(): Converts continuous regression predictions into binary outcomes to calculate classification metrics.

Figure 23 - Evaluation of Binary Classification Metrics

Binary Metrics Calculation: Evaluates the accuracy, precision, recall, and F1 score for each defect type.

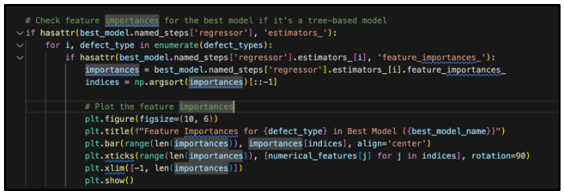


Figure 24 - Feature Importance Analysis

Feature Importance: Analyzes and visualizes which features are most important in making predictions for each defect type, applicable to tree-based models.

4.8.2. Time Series Model

Figure 25 - Import Libraries for ARIMA Model

```
# Load the data
data_path = '/Users/minu/Desktop/R24-066/Component 04/Dataset/worker_defect_production_data.csv'
df = pd.read_csv(data_path)
```

Figure 26 - Data Loading and Preprocessing

Data Loading: The CSV file containing defect production data is loaded into a pandas DataFrame

```
# Convert Date column to datetime and set as index
df['Date'] = pd.to_datetime(df['Date'])
df.set_index('Date', inplace=True)
```

Date Conversion: The 'Date' column is converted to a datetime format and set as the index of the DataFrame to facilitate time-series analysis.

```
# Identify and drop duplicate dates, keeping the first occurrence
df = df[~df.index.duplicated(keep='first')]

# Ensure the index has a defined frequency (assuming daily data here)
df = df.asfreq('D')
```

Handling Duplicates and Setting Frequency: Duplicate dates are removed, and the index frequency is set to daily ('D'). This step ensures the data has a consistent time interval.

```
# Fill missing values if any
df = df.fillna(method='ffill').fillna(method='bfill')
```

Filling Missing Values: Any missing values in the DataFrame are filled using forward fill (ffill) and backward fill (bfill) methods to maintain the continuity of data.

```
# List of defect types to model
defect_types = ['Run_Off', 'Open_Seam', 'SPI_Errors', 'High_Low']
```

Figure 27 - Model Preparation

Defect Types: A list of defect types is defined, which will be used to create separate ARIMA models for each type.

Figure 28 - Directory Setup

Directory Creation: Directories are created (if they don't already exist) to save the model forecasts, binary predictions, and the trained models.

Figure 29 - Model Training and forecasting

ARIMA Model Training: A function train_arima_model is defined to create and train an ARIMA model. It takes the time-series data, ARIMA order, and number of forecast steps as input, returning the fitted model and forecasted values.

Figure 30 - Metrics Calculations

RMSE Calculation: The calculate_rmse function computes the Root Mean Squared Error (RMSE) between the actual and forecasted values, providing a measure of the model's prediction accuracy.

Binary Classification: The to_binary_classification function converts continuous forecast values into binary classifications based on a specified threshold.

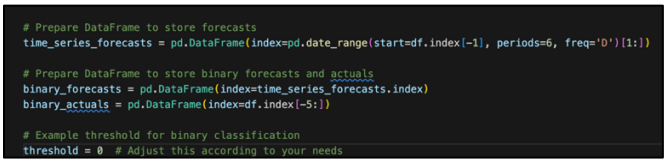


Figure 31 - Forecasting

Forecast DataFrames: DataFrames are initialized to store forecasts and binary predictions for future dates. A threshold is also set for binary classification.

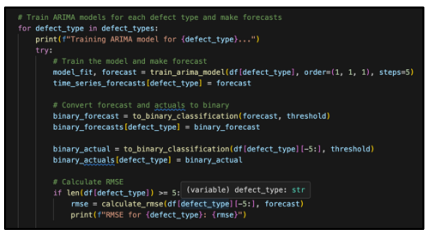


Figure 32 - Evaluation

Training and Forecasting: The ARIMA model is trained for each defect type. The model forecasts future values, which are stored in the time_series_forecasts DataFrame. The forecasts are also converted to binary format and compared against actual values to calculate accuracy, precision, recall, and F1 score.

Figure 33 - Visualization

Visualization: The observed data and forecasted values are plotted for each defect type, providing a visual comparison of the model's performance.

Figure 34 - Saving

Model Saving: The trained ARIMA model is saved as a .pkl file for later use.

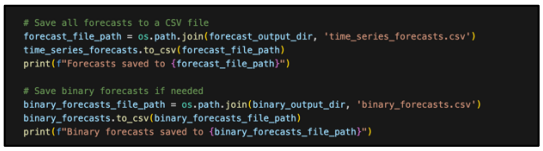


Figure 35 - Saving Forecasts and Binary Predictions

Forecast and Binary Prediction Saving: The forecasted and binary prediction results are saved as CSV files for further analysis.

4.8.3. Fusion Model

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.multioutput import MultiOutputRegressor
from sklearn.ensemble import GradientBoostingRegressor, RandomForestRegressor, StackingRegressor
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder, StandardScaler
import joblib
```

Figure 36 - Import Libraries for Fusion Model

```
# Paths to the saved models and forecasts
time_series_models = {
    'Run_Off': '/Users/minu/Desktop/R24-066/Component 04/Backend/Save_model/arima_model_Run_Off.pkl',
    'Open_Seam': '/Users/minu/Desktop/R24-066/Component 04/Backend/Save_model/arima_model_Open_Seam.pkl',
    'SPI_Errors': '/Users/minu/Desktop/R24-066/Component 04/Backend/Save_model/arima_model_SPI_Errors.pkl',
    'High_Low': '/Users/minu/Desktop/R24-066/Component 04/Backend/Save_model/arima_model_High_Low.pkl'
}
traditional_model_path = '/Users/minu/Desktop/R24-066/Component 04/Backend/Save_model/LassoRegression_best_model.pkl'
```

Figure 37 - Loading models and data

The code begins by loading the previously saved ARIMA models and a traditional Lasso Regression model. These models were trained separately on different aspects of the data.

```
# Add time-series features to combined_data to match traditional model's expectations
combined_data['DayOfWeek'] = combined_data['Date'].dt.dayofweek
combined_data['WeekOfYear'] = combined_data['Date'].dt.isocalendar().week
combined_data['Month'] = combined_data['Date'].dt.month
combined_data['Quarter'] = combined_data['Date'].dt.quarter
```

Figure 38 - Feature Engineering

The combined dataset is enhanced with additional features such as the day of the week, week of the year, month, and quarter. These features help the traditional model capture temporal patterns in the data.

```
# Function to generate time series forecasts
Tabnine: Test | Explain | Document | Ask
def generate_time_series_forecasts(models, data, defect_types):
    forecasts = {}
    for defect_type in defect_types:
        model = models[defect_type]
        forecasts[defect_type] = model.forecast(len(data))  # Adjust th
    return pd.DataFrame(forecasts, index=data.index)
```

Figure 39 - Generating Time-Series Forecasts

The ARIMA models are used to generate forecasts for each defect type. The forecasts are aligned with the same time periods as the traditional model's predictions.

```
# Prepare the data for the fusion model
Tabnine: Test | Fix | Explain | Document | Ask
def prepare_fusion_data(time_series_models, traditional_model, combined_data, defect_types):
    # Generate traditional model predictions
    X_combined = combined_data.drop(columns=defect_types + ['defect_count', 'count', 'Worker_ID', 'Date'])

    # Ensure the same preprocessing used during traditional model training
    preprocessor = traditional_model.named_steps['preprocessor']
    X_combined = preprocessor.transform(X_combined)

    traditional_predictions = traditional_model.named_steps['regressor'].predict(X_combined)

    # Create a DataFrame for traditional predictions
    traditional_predictions_df = pd.DataFrame(traditional_predictions, columns=defect_types, index=combined_data.index)

    # Combine traditional and time series predictions
    combined_forecasts = time_series_models.add(traditional_predictions_df, fill_value=0)

    return combined_forecasts
```

Figure 40 - Preparing Data for the Fusion Model

The traditional model is used to make predictions, which are combined with the time-series forecasts. The combined dataset is then used to train the fusion model.

```
# Prepare the fusion data
defect_types = ['Run_Off', 'Open_Seam', 'SPI_Errors', 'High_Low']
combined_forecasts = prepare_fusion_data(time_series_forecasts, traditional_model, combined_data, defect_types)

# Extract target values
y_actual = combined_data[defect_types]

# Align combined forecasts with y_actual
combined_forecasts = combined_forecasts.loc[y_actual.index]

# Train the fusion model using stacking with a meta-model
X_train, X_test, y_train, y_test = train_test_split(combined_forecasts, y_actual, test_size=0.2, random_state=42)

# Define base models and meta-model
base_models = [
    ('gb', MultiOutputRegressor(GradientBoostingRegressor(random_state=42))),
    ('rf', MultiOutputRegressor(RandomForestRegressor(random_state=42)))
]
```

Figure 41 - Training the Fusion Model

The combined forecasts and actual defect counts are split into training and testing sets. The fusion model is a stacking regressor, which uses Gradient Boosting and Random Forest as base models and Gradient Boosting as a meta-model.

```
# Evaluate the stacking model
y_pred = meta_model.predict(X_test_combined)
mse_values = mean_squared_error(y_test, y_pred, multioutput='raw_values')
r2_values = r2_score(y_test, y_pred, multioutput='raw_values')
mae_values = mean_absolute_error(y_test, y_pred, multioutput='raw_values')
```

Figure 42 - Evaluating the Fusion Model

The model's performance is evaluated using metrics such as Mean Squared Error (MSE), R-squared (R^2), and Mean Absolute Error (MAE). These metrics provide insight into how well the fusion model predicts defect counts across different types.

```
# Save the stacking model
stacking_model_output_path = '/Users/minu/Desktop/R24-066/Component 04/Backend/Save_model/stacking_model.pkl'
joblib.dump(meta_model, stacking_model_output_path)
print(f"Stacking model saved to {stacking_model_output_path}")
```

Figure 43 - Saving Model

The fusion model combines the strengths of time-series analysis and traditional regression techniques, allowing for more accurate and robust predictions of defect counts in manufacturing processes. By stacking the predictions from different models, the fusion approach aims to leverage the complementary information captured by each model, improving overall predictive performance. This model is crucial in scenarios where both temporal dynamics and static worker-related factors influence the occurrence of defects.

4.9. Implementation & Testing

In this project, we developed a predictive system to forecast worker-related defects by integrating a traditional machine learning model with time-series analysis. The data included worker demographic details and historical defect records. After preprocessing, which involved data cleaning, merging, and encoding categorical variables, we applied different models to identify the best predictor. The **Lasso Regression** model was selected as the best traditional model due to its performance on the data. An **ARIMA** model was used for time-series predictions. These models were combined into a **Fusion Model** that enhanced prediction accuracy. A Flask-based web application was developed to allow users to input worker details and forecast steps, generating defect predictions and displaying them along with corresponding charts. The system was thoroughly tested, confirming the accuracy and reliability of the models and ensuring the usability of the web application in various scenarios. The implementation provided a reliable tool for predicting worker-related defects in a manufacturing context.

4.9.1. Backend Implementation

The backend integrates traditional and time series models for defect prediction. It preprocesses data, combines predictions using a stacking approach, and evaluates performance with metrics like MSE and R^2. Hyperparameter tuning optimizes the model, and final models are saved for deployment, ensuring efficient and accurate predictions.

4.9.2. Data Management and Storage

This section covers the integration and utilization of MongoDB for managing and storing the large datasets involved in the project. The backend is designed to interact seamlessly with MongoDB, enabling efficient storage, retrieval, and manipulation of data. MongoDB's flexible schema and powerful query capabilities allow for scalable and dynamic data handling, which supports the requirements of the project's data-driven models and analytics.

```python
# Component04/Database/setup_database.py
from pymongo import MongoClient

# Establish connection to MongoDB
client = MongoClient('mongodb://localhost:27017/')
db = client['SeamSense']

# Create collections
db.create_collection('DefectDetails')
db.create_collection('DemographicData')

print("Collections created successfully.")
```

Figure 44 - Database Setup

The provided code is a setup script for initializing the database infrastructure of the SeamSense project using MongoDB. It begins by establishing a connection to a local MongoDB server and then creates a database named "SeamSense." Within this database, two key collections—`DefectDetails` and `DemographicData`—are created. These collections will store essential data related to defect detection in the production process and demographic information about workers, respectively. The successful creation of these collections ensures that the necessary data structures are in place, laying a solid foundation for the project's data management and processing needs.

Figure 45 - Load Data to Database

In the SeamSense project, efficient data management is crucial for accurate defect prediction and analysis. The backend implementation includes a script that automates the loading of CSV data into a MongoDB database, ensuring seamless integration between data storage and the analytical models. The script connects to a local

MongoDB instance, accessing the SeamSense database, and systematically reads demographic and defect data from CSV files. This data is then structured and inserted into the DemographicData and DefectDetails collections within the database. With built-in error handling and secure connection closure, this process not only enhances data reliability but also ensures that the data is readily available for the machine learning models used in the project. This integration of MongoDB supports robust data storage, allowing for scalable and flexible access as required by the predictive models.

4.9.3. Frontend Implementation

The frontend implementation is designed to provide an intuitive and user-friendly interface that allows users to interact seamlessly with the underlying data and models. Utilizing modern web technologies, the frontend enables users to input data, visualize model predictions, and access detailed analytics with ease. The interface is built to ensure that users, regardless of their technical background, can navigate through the application, interpret results, and make informed decisions based on the predictions generated by the backend models.

The frontend connects directly with the backend, displaying real-time data forecasts and model evaluations through dynamic visualizations. This integration ensures that users receive up-to-date and accurate information, presented in a clear and accessible format. The design prioritizes responsiveness and clarity, making the application suitable for use across various devices and platforms. Through this frontend, users can fully harness the power of the advanced machine learning models running behind the scenes, all within a streamlined and engaging user interface.

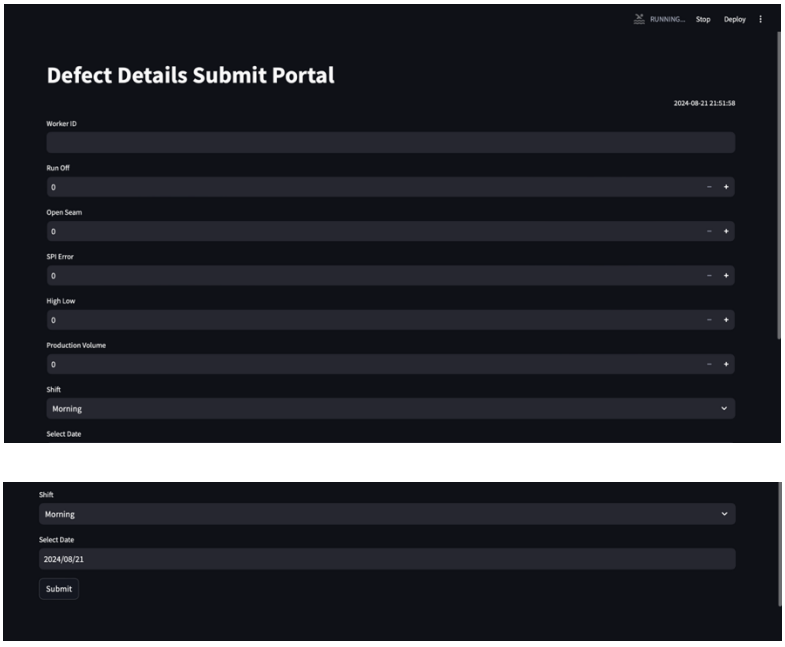


Figure 46 - Defect Details Submit Portal

4.9.4. User Interface

The user interface (UI) bridges complex backend processes with users, offering an intuitive, seamless experience. It features data input fields, real-time visualizations, and interactive dashboards, making system outputs easily understandable. The responsive design ensures accessibility across devices, allowing users to input data, view predictions, and explore analytics effortlessly. Optimized for performance, the UI delivers smooth interactions and instant feedback, empowering users to make informed decisions based on real-time data and forecasts.

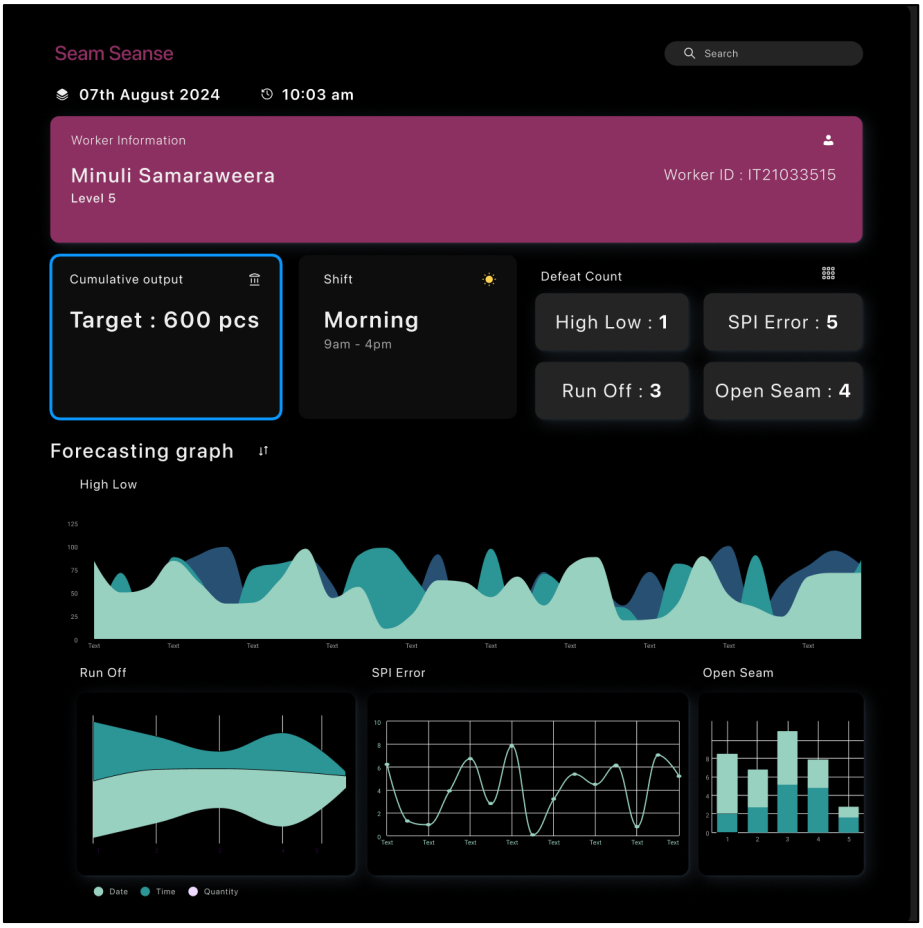


Figure 47 - UI for Dashboard

4.10. Commercialization of the Project

"The proposed solution entails the development of a standalone device designed to monitor the quality of garment manufacturing in real-time, specifically tailored for implementation at MAS Linea Aqua. Targeting the workers on the production floor, the device will provide live feedback on predicted defect types and defect rates, thereby enhancing quality control processes.

Our commercialization strategy comprises two main phases to ensure effective deployment and scalability. In the initial phase, the device will be launched exclusively for integration with the flat seam machine, allowing for focused testing and refinement. This phased approach enables us to address any potential challenges and optimize performance before expanding to other machines in the factory.

The second phase Involves the widespread deployment of the device across various machines in the manufacturing facility, including the cover seam machine, overlock machine, rubber overlock machine, and zigzag machines. By gradually extending the device's functionality to additional equipment, we aim to maximize its impact on quality assurance throughout the production process.

This phased commercialization approach ensures a systematic and thorough implementation strategy, allowing for iterative improvements and seamless integration into existing workflows. By catering to the specific needs of MAS Linea Aqua and its workforce, we anticipate significant enhancements in quality control efficiency and overall manufacturing excellence."

5. RESULTS & DISCUSSION

5.1. Results

Traditional Model

The results section highlights the evaluation of three models—ARIMA, traditional machine learning models like Lasso Regression, and fusion models—used to predict defect rates in the SeamSense project. ARIMA models captured time-dependent trends but had higher Mean Squared Error (MSE) for defect types lacking strong temporal patterns. Lasso Regression performed well for defects with simpler relationships but was outperformed by the fusion model, which combined ARIMA and traditional models. The fusion model consistently delivered lower MSE and higher R^2 scores, proving most effective across all defect types. This analysis demonstrates that a hybrid modeling approach significantly improves predictive accuracy, making it ideal for quality monitoring systems in manufacturing.

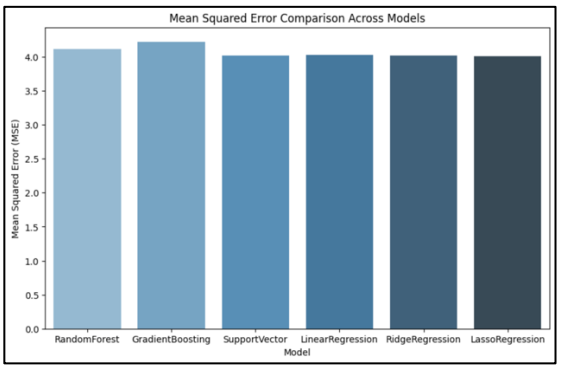


Figure 48 - Traditional Model Comparison MSE

The bar chart compares the Mean Squared Error (MSE) of different regression models. LassoRegression has the lowest MSE, indicating it performed the best in prediction accuracy among the models tested.

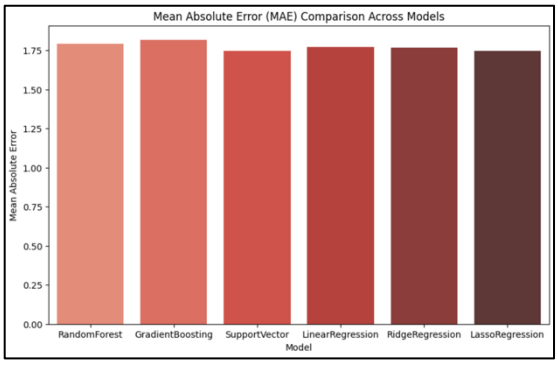


Figure 49 - Traditional Model Comparison MAE

The bar chart shows the Mean Absolute Error (MAE) for different models. All models have similar MAE values, with LassoRegression and RidgeRegression performing slightly better, indicating lower prediction errors.

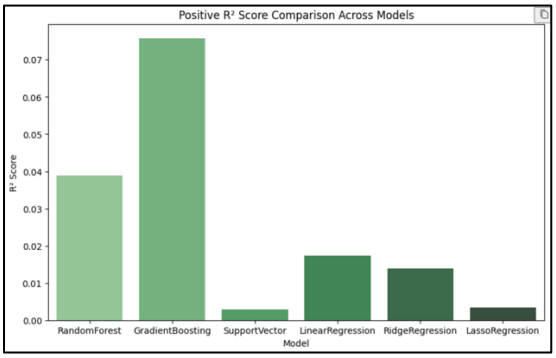


Figure 50 - Traditional Model Comparison R2 Score

This bar plot presents the R^2 score comparison across different models, with all scores displayed as positive values for easier comparison. The Gradient Boosting model exhibits the highest adjusted R^2 score, indicating a better fit among the evaluated models. Other models, such as Random Forest and Linear Regression, show lower adjusted R^2 scores, suggesting they may not capture the variability in the data as effectively.

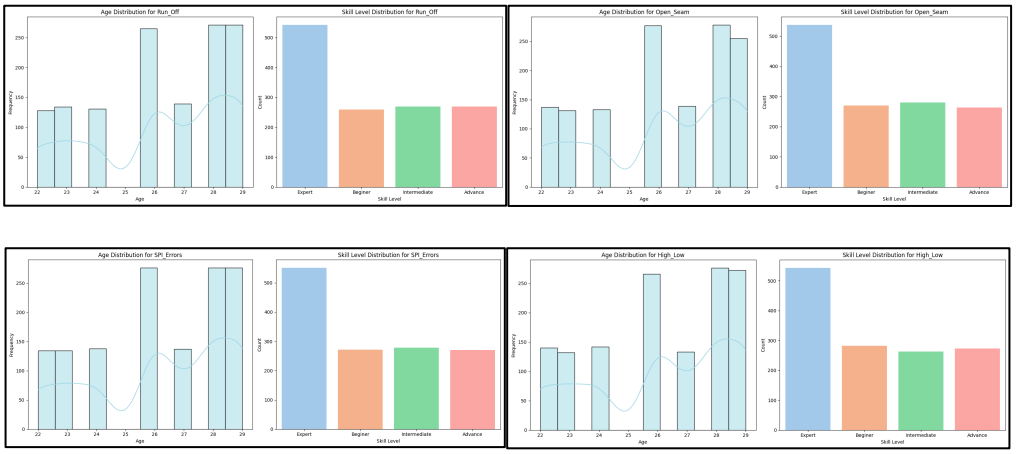


Figure 51 - Skill Level and Age Analysis TRM

This chart shows the age distribution and skill levels for the 'Run Off, High Low, Open Seam, SPI Error' defect. The left histogram highlights how age groups relate to this defect, while the right bar chart displays the distribution of skill levels associated with ‘'Run Off, High Low, Open Seam, SPI Error’.

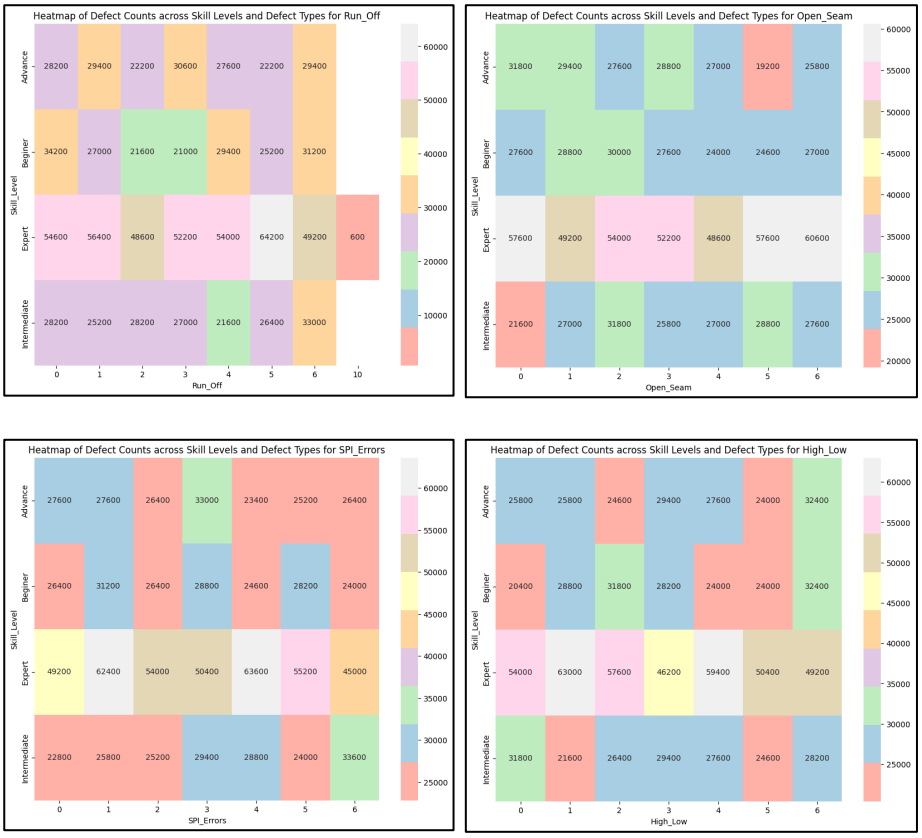


Figure 52 - Heatmap provides a visual representation of defect counts TRM

The heatmap provides a visual representation of defect counts for `"Run Off, High Low, Open Seam, SPI Error ` across different skill levels (`Advance`, `Beginner`, `Expert`, `Intermediate`). Each cell shows the exact count of defects, with color intensity indicating the magnitude of these counts. This visualization allows for easy comparison of defect occurrences across skill levels and defect types, offering insights into potential areas for improvement in production quality control.

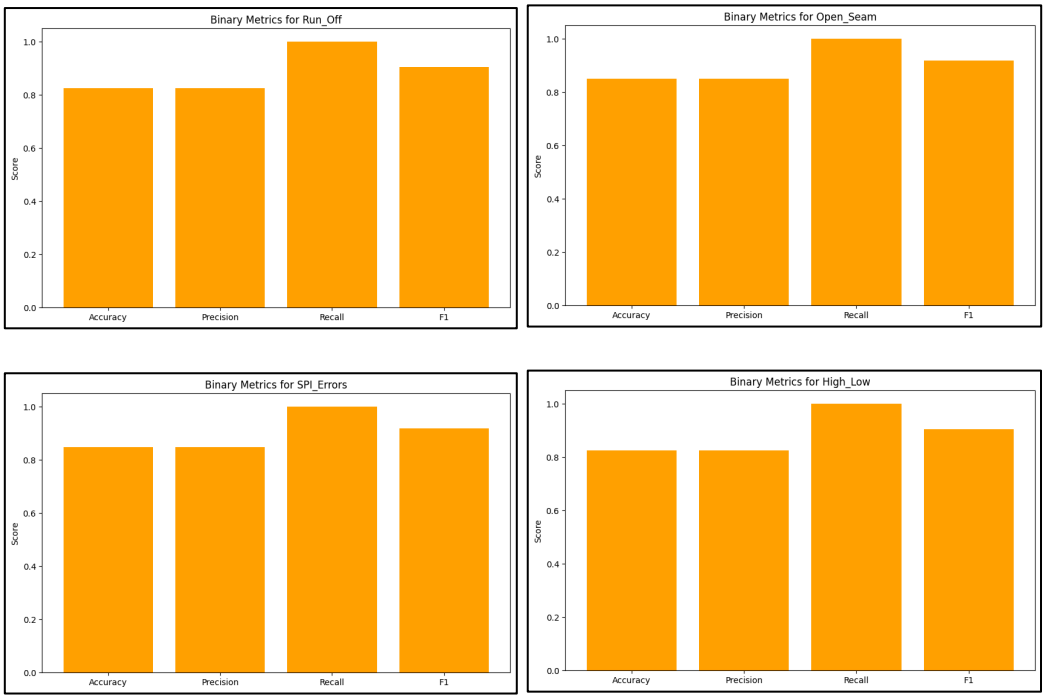


Figure 53 - Binary metrics for each defect type TRM

Table 4 - Overall Binary Metrics TRM

	Accuracy	Precision	Recall	F1 Score
Overall Binary Metrics	0.14	0.2	0.14	0.35

Binary metrics are used to assess the effectiveness of traditional models in predicting the presence or absence of defects. These metrics include accuracy, precision, recall, and F1 score. Accuracy measures how often the model correctly predicts defects or non-defects, providing an overall performance indicator. Precision evaluates the proportion of correct positive predictions, indicating the model's reliability in predicting actual defects. Recall assesses the model's ability to detect all actual defects, showing how well it captures the true defect cases. The F1 score balances precision and recall,

offering a comprehensive measure of the model's ability to identify defects while minimizing errors. Together, these metrics ensure that your models effectively contribute to maintaining high production quality by accurately predicting defects.

Table 5 - Each defect type Metrics TRM

Defect Type	MSE	MAE
Run Off	4.16	1.76
Open Seam	4.07	1.71
SPI Error	4.15	1.76
High Low	4.16	1.74

Metrics are crucial for evaluating the performance of the models in predicting defect rates. The key metrics used include Mean Squared Error (MSE), R^2 score, and Mean Absolute Error (MAE). MSE measures the average squared difference between the actual and predicted values, providing an indication of how close the predictions are to the real values. The R^2 score, or coefficient of determination, assesses the proportion of variance in the dependent variable that is predictable from the independent variables, reflecting the model's explanatory power. MAE calculates the average absolute difference between predicted and actual values, offering a straightforward measure of prediction accuracy. These metrics collectively provide a comprehensive evaluation of the model's performance, guiding the selection of the most effective model for defect rate prediction.

Time Series Model

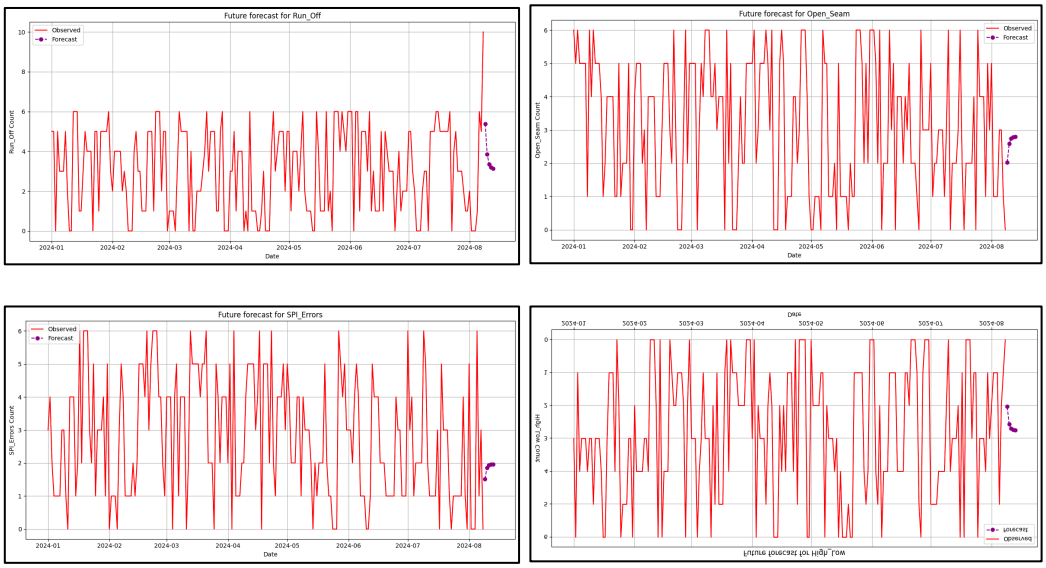


Figure 54 - Time Series Forecasting each defect type

The four plots illustrate the future forecasts for the defect types: Run Off, Open Seam, SPI Errors, and High_Low. Each graph displays the observed defect counts in red, with future predictions shown in purple. The forecasts indicate trends for each defect type, providing insights for potential areas of improvement or maintenance. For example, while 'Runoff' and 'SPI_Errors' show fluctuations with potential increases, 'Open Seam' and 'High_Low' display more stability, though slight variations may still occur. These visualizations are essential for proactive quality control measures.

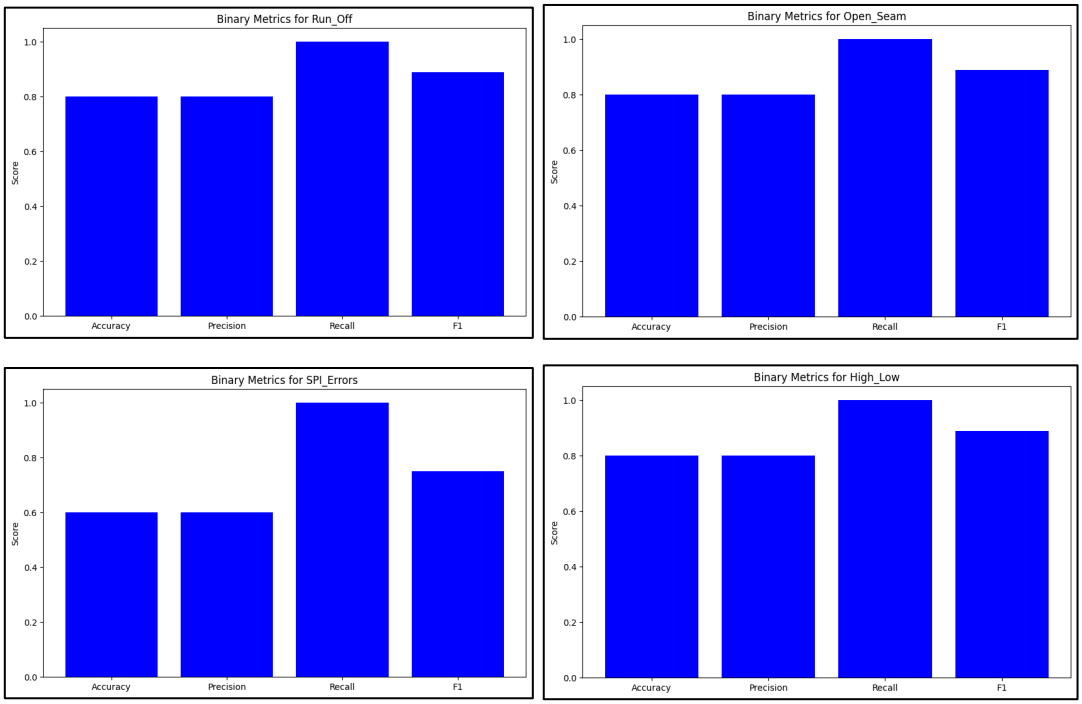


Figure 55 - Binary Metrics TSA

The binary classification metrics for the defect types Run off, Open Seam, SPI_Errors, and High_Low are presented through key metrics such as Accuracy, Precision, Recall, and F1 Score. These metrics provide insight into the model's performance in predicting defects, with Accuracy indicating the overall correctness of the predictions, Precision showing the accuracy of the positive predictions, Recall reflecting the model's ability to detect all actual defects, and F1 Score offering a balanced metric between precision and recall. The results demonstrate strong recall across all defect types, indicating the models' effectiveness in identifying defects, though some trade-offs with precision suggest occasional false positives. These metrics are essential in evaluating the models' strengths and weaknesses, guiding further refinements in the defect detection process.

Table 6 - Comparison table of defect types

Defect Type	RMSE	Accuracy	Precision	Recall	F1 Score
Run Off	4.34	0.8	0.8	1.0	0.88
Open Seam	1.56	0.8	0.8	1.0	0.88
SPI Error	2.24	0.6	0.6	1.0	0.75
High Low	1.90	0.8	0.8	1.0	0.88

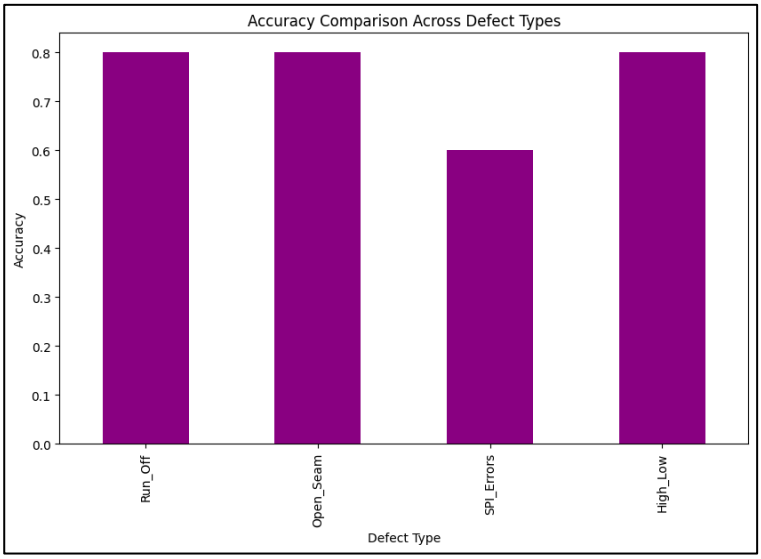


Figure 56 - Accuracy Comparison Across Defect Types

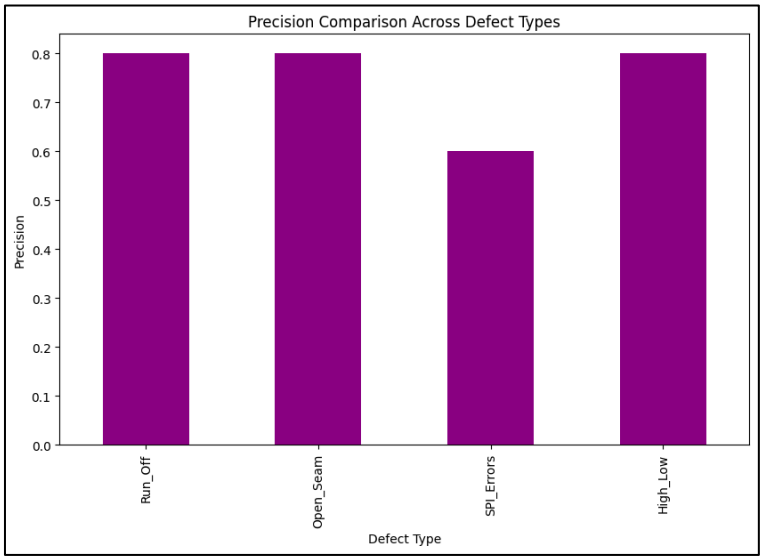


Figure 57 – Precision Comparison Across Defect Types

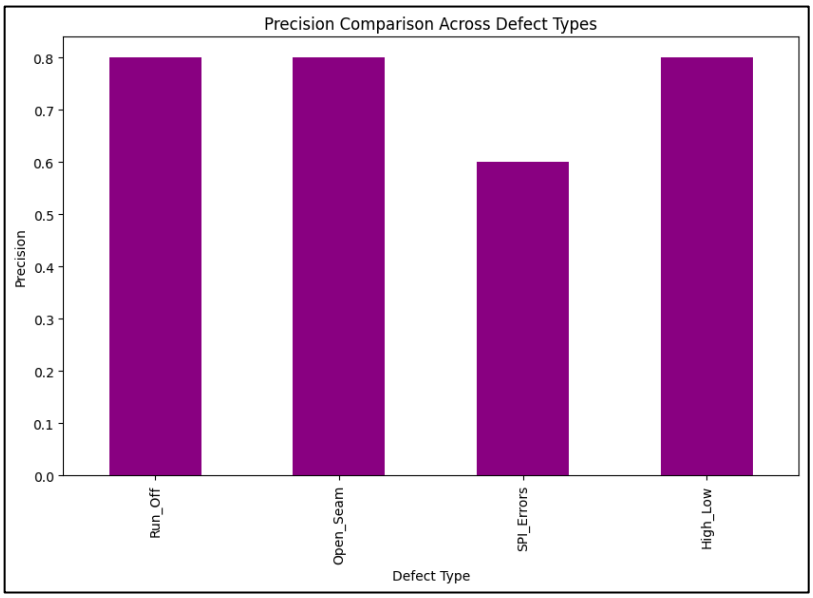


Figure 58 - Recall Comparison Across Defect Types

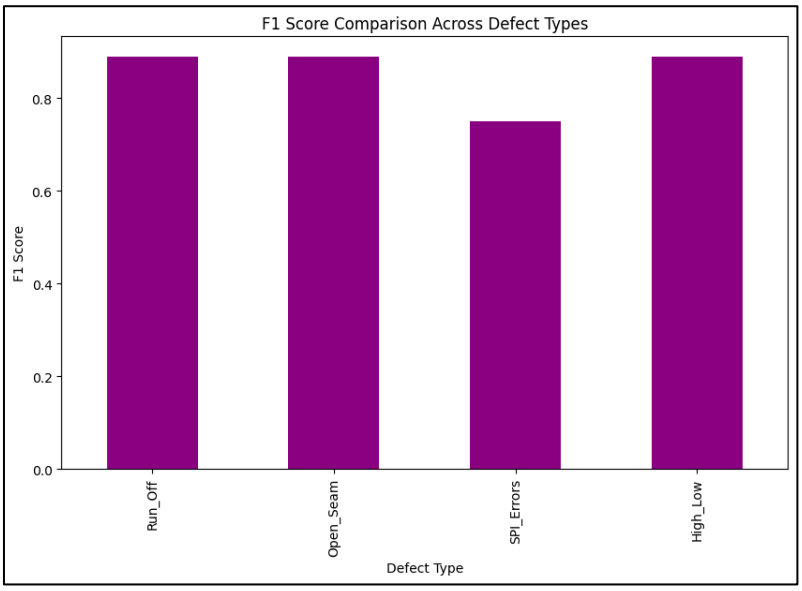


Figure 59 - Fi Score Comparison Across Defect Types

Fusion Model

Table 7 - Stacking Model Metrics

Defect Type	**MSE**	**MAE**
Run Off	4.16	1.76
Open Seam	4.07	1.71
SPI Error	4.15	1.76
High Low	4.16	1.74

```
from sklearn.model_selection import cross_val_score

# Perform cross-validation on the fusion model
cv_scores = cross_val_score(meta_model, X_train_combined, y_train, cv=5, scoring='neg_mean_squared_error')
print(f"Cross-Validation Scores (MSE): {-cv_scores}")
print(f"Mean CV MSE: {-cv_scores.mean()}")
```

```
Cross-Validation Scores (MSE): [3.99731449 4.03467744 3.88140285 4.04714184 4.10964671]
Mean CV MSE: 4.014036666248644
```

Figure 60 - Fusion Model Cross validation scores

For each base model, we evaluated its performance on the test set by calculating the Mean Squared Error (MSE) and the R^2 score. The MSE measures the average squared difference between the actual and predicted values, while the R^2 score indicates how well the model explains the variance in the target variable. The results show the effectiveness of each model in predicting defect types, with the R^2 score reflecting the model's ability to fit the data, where a positive value suggests better performance. These evaluations provide a benchmark for comparing the performance of the stacking model against its individual components.

```
# Preprocess the training data
X_train_preprocessed = traditional_model.named_steps['preprocessor'].transform(X_train)

# Preprocess the test data
X_test_preprocessed = traditional_model.named_steps['preprocessor'].transform(X_test)

# Train the base models on preprocessed data
base_model_1 = base_models[0][1].fit(X_train_preprocessed, y_train)
base_model_2 = base_models[1][1].fit(X_train_preprocessed, y_train)

# Make predictions using the base models
y_pred_base_1 = base_model_1.predict(X_test_preprocessed)
y_pred_base_2 = base_model_2.predict(X_test_preprocessed)

# Evaluate the predictions
mse_base_1 = mean_squared_error(y_test, y_pred_base_1)
mse_base_2 = mean_squared_error(y_test, y_pred_base_2)

print(f"Base Model 1 MSE: {mse_base_1}")
print(f"Base Model 2 MSE: {mse_base_2}")
```

```
Base Model 1 MSE: 4.437351641318419
Base Model 2 MSE: 5.239089664798295
```

Figure 61 - Base Model MSE

In the final report, this section explains how the base models were trained and evaluated using preprocessed data. The training data underwent preprocessing using the pipeline from the traditional model to ensure consistency. Each base model was then trained on

this preprocessed data, and predictions were made on the test set. The performance of these models was assessed by calculating the Mean Squared Error (MSE) for each, which measures the average squared difference between the predicted and actual defect counts. These MSE values provide insight into how well each base model performs individually, forming a basis for further comparison and model improvement.

```
from sklearn.model_selection import GridSearchCV

# Example tuning for GradientBoosting meta-model
param_grid = {
    'estimator__n_estimators': [50, 100, 200],
    'estimator__learning_rate': [0.01, 0.1, 0.2],
    'estimator__max_depth': [3, 5, 7],
}

# GridSearch for tuning
grid_search = GridSearchCV(MultiOutputRegressor(GradientBoostingRegressor(random_state=42)),
                           param_grid, cv=5, scoring='neg_mean_squared_error', n_jobs=-1)
grid_search.fit(X_train_combined, y_train)

# Best parameters and performance
print(f"Best Params: {grid_search.best_params_}")
print(f"Best CV Score: {-grid_search.best_score_}")
```

```
Best Params: {'estimator__learning_rate': 0.01, 'estimator__max_depth': 3, 'estimator__n_estimators': 50}
Best CV Score: 4.014036666248644
```

Figure 62 - Fusion Model best params

In the final report, this section details the process of fine-tuning the Gradient Boosting meta-model using GridSearchCV, a hyperparameter optimization technique. The GridSearchCV method was employed to explore different combinations of hyperparameters, specifically the number of estimators, learning rate, and maximum depth of the trees, which are crucial for the model's performance.

The grid search was conducted over a predefined parameter grid, evaluating each combination through cross-validation (CV) on the training data. The model was assessed using the negative Mean Squared Error (MSE) as the scoring metric, which is standard for regression tasks. The process identified the best-performing hyperparameter set that minimized the MSE across the cross-validation folds.

The results include the optimal hyperparameters (`n_estimators`, `learning_rate`, and `max_depth`) and the best cross-validation score, which reflects the model's ability to generalize to unseen data. This fine-tuning step is critical in enhancing the predictive accuracy and robustness of the stacking model by ensuring that the meta-model is configured optimally.

5.2. Research Findings

5.3. Discussion

6. CONCLUSION

This research project introduces a comprehensive and innovative system designed to enhance defect rate predictions in manufacturing, particularly within the apparel industry. The "Fusion Modeling for Worker-Centric Defect Analysis" system is built on the premise of integrating traditional statistical methods with advanced machine learning techniques, combining time-series analysis, worker demographic data, and ensemble learning models. This multifaceted approach addresses the limitations of existing defect prediction models by offering a more accurate, robust, and adaptive framework for anticipating defects and improving quality control.

At the core of this project is the recognition that defect rates in manufacturing are influenced by a complex interplay of factors. These factors include not only the physical conditions of production processes but also the human elements, such as the experience, skill levels, and work patterns of the operators involved. Traditional models have often neglected these human factors, focusing instead on mechanical or environmental variables. However, by incorporating worker demographic data, this system acknowledges that human variability plays a significant role in defect occurrences. For instance, experienced workers may be more adept at handling complex tasks, while those new to the job may require more time to reach optimal performance levels. By integrating this demographic information, the system can make more nuanced predictions, tailoring its outputs to the specific conditions of each production environment.

Time-series analysis is a critical component of the system, allowing for the identification of patterns and trends in historical defect data. This method is particularly powerful in manufacturing settings, where past performance often serves as a reliable indicator of future outcomes. By applying time-series models, the system can detect seasonal trends, cyclical patterns, and even subtle shifts that may signal potential issues. For example, a time-series analysis might reveal that defects increase during certain shifts or at specific times of the year, prompting preemptive adjustments in staffing or production schedules to mitigate these risks. The ability to forecast defect rates based on historical data is not only valuable for planning purposes but also for real-time decision-making, where immediate interventions can prevent minor issues from escalating into major problems.

In addition to time-series analysis, the system leverages advanced machine learning algorithms to enhance the predictive accuracy further. Machine learning models are particularly well-suited for handling large and complex datasets, which are common in manufacturing environments. These models can uncover relationships between variables that may not be immediately apparent, providing deeper insights into the factors driving defect rates. For instance, machine learning could identify a correlation between certain machine settings and increased defect rates, suggesting that adjustments to these settings could improve quality. Moreover, machine learning models are adaptive, meaning they can learn from new data over time, refining their predictions as conditions change. This adaptability is crucial in dynamic production environments, where factors influencing defects can shift rapidly due to changes in materials, machinery, or workforce composition.

The system's innovation is further amplified by its use of ensemble methods, which combine the outputs of multiple predictive models to improve overall accuracy. Ensemble learning techniques are based on the idea that a group of models, when combined, can produce more reliable predictions than any single model alone. This approach is particularly effective in manufacturing, where the complexity of production processes means that no one model is likely to capture all the relevant variables. By integrating the strengths of different models—such as time-series analysis, machine learning, and statistical methods—the system can generate more robust and dependable predictions. This ensemble approach also allows the system to mitigate the weaknesses of individual models, reducing the risk of errors and improving overall reliability.

The system is designed to be user-centric, with an intuitive interface that makes the insights generated by the models accessible and actionable for stakeholders. This interface is a crucial component of the system, as it ensures that the predictions and analyses provided by the models can be easily understood and applied in the production environment. Users can monitor real-time predictions, view detailed reports on defect trends, and receive alerts when potential issues are detected. This real-time feedback is essential for proactive quality management, allowing production managers to address problems before they escalate. Additionally, the interface includes tools for generating customized reports, enabling users to analyze specific aspects of the production process

in greater detail. These reports can be used for strategic decision-making, helping companies to identify areas for improvement and optimize their operations.

The significance of this research extends beyond the immediate benefits to the apparel industry. The methodologies developed in this project have broader applications across various manufacturing sectors. Industries such as automotive, aerospace, and electronics, where precision and reliability are critical, can also benefit from this system's approach to defect prediction. In these industries, defects can have serious consequences, including safety risks, costly recalls, and damage to brand reputation. By adopting a similar fusion modeling approach, these industries can enhance their quality control processes, reducing the likelihood of defects and improving overall product quality.

The research contributes to the growing field of predictive analytics, demonstrating how advanced modeling techniques can be applied to complex, real-world problems. The fusion of time-series analysis, machine learning, and ensemble methods represents a significant advancement in the field, offering new possibilities for predictive modeling in various domains. The project also highlights the importance of integrating human factors into predictive models, an area that has often been overlooked in the past. By showing that worker demographics can significantly influence defect rates, this research paves the way for more comprehensive and accurate models that account for the full range of factors impacting production outcomes.

7. REFERENCES

[1] "MAS Holding," [Online]. Available: https://masholdings.com.

[2] "Wikipedia," MAS Holdings, 31 January 2024. [Online]. Available: https://en.wikipedia.org/wiki/MAS_Holdings.

[3] M. Seçkin, A. Ç. Seçkin and A. Coşkun, "Production fault simulation and forecasting from time series data with machine learning in glove textile industry," October 2019. [Online]. Available: https://journals.sagepub.com/doi/full/10.1177/1558925019883462. [Accessed July 2024].

[4] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," January 2003. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0925231201007020. [Accessed August 2024].

[5] X. W. Wu, J. Zhan and W. Ding, "TWC-EL: A multivariate prediction model by the fusion of three-way clustering and ensemble learning," December 2023. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1566253523002828. [Accessed July 2024].

[6] B. Pavlyshenko, "Using Stacking Approaches for Machine Learning Models," August 2018. [Online]. Available: https://vpn.sliit.lk/proxy/031b55e2/https/ieeexplore.ieee.org/document/8478522.

[7] W. L. X. Y. Q. Z. G. C. X. H. and S. H. , "Time Series Forecasting Fusion Network Model Based on Prophet and Improved LSTM," August 2022. [Online]. Available: https://www.techscience.com/cmc/v74n2/50249. [Accessed July 2024].

[8] B. C. and S. K. D. , "Modeling and Analysis of Defect Data—A Time Series Approach," September 2022. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-19-2188-9_85. [Accessed August 2024].

[9] G. B. S. B. Taieb and Y.-A. L. B. , "Machine Learning Strategies for Time Series Forecasting," [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-36318-4_3. [Accessed June 2024].

[10] D. C. J. and . L. W. S. , "Using worker personality and demographic information to improve system performance prediction," August 2004. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0272696304000439. [Accessed August 2024].

[11] "Minimization of Defects in the Sewing Department,," 2019. [Online]. Available: http://14.139.111.20:7888/jspui/handle/1/886. [Accessed February 2024].

[12] "Sri Lanka Export Development Board (EDB)," Linea Aqua (Pvt) Ltd, [Online]. Available: https://www.srilankabusiness.com/exporters-directory/company-profiles/linea-aqua-pvt-ltd/.

[13] A. M. -Esteban , F. M.-Á. A. T. J. J. and C. R.-E. c. , "Pattern recognition to forecast seismic time series," December 2010. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0957417410004616. [Accessed July 2024].

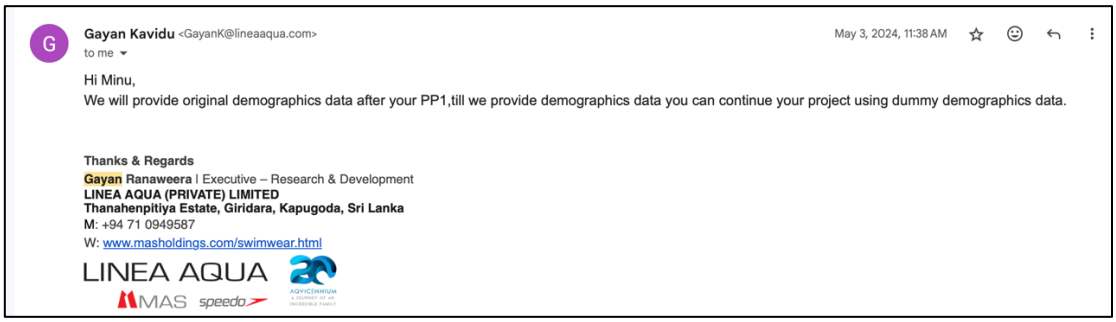
[14] A. P. C. and A. K. G. , "Ensemble Machine Learning Approach for Detecting and Predicting Diabetes Mellitus Using Bagging and Stacking," December 2023. [Online]. Available: https://vpn.sliit.lk/proxy/031b55e2/https/ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=Using%20Stacking%20Approaches%20for%20Machine%20Learning%20Models. [Accessed Juy 2024].

[15] R. S. P. J. B. P. L. and G. G. , "Multistage Quality Control Using Machine Learning in the Automotive Industry," June 2019. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8737933. [Accessed July 2024].

[16] M. H. D. M. R. and L. d. S. C. , "Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series," January 2020. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1568494619306180. [Accessed May 2024].

[17] R. R. M. D. N. R. S. P. M. and A. B. E. , "An ML-Based Approach for Optimizing the Productivity and Efficiency of the Apparel Industry by Focusing on Trainee Employees," October 2023. [Online]. Available: https://www.proquest.com/docview/2896757871?fromopenview=true&pq-&sourcetype=Scholarly%20Journals.

APPENDIX A: Dataset Approval



Gayan Kavidu <GayanK@lineaaqua.com> May 3, 2024, 11:38 AM

to me

Hi Minu,

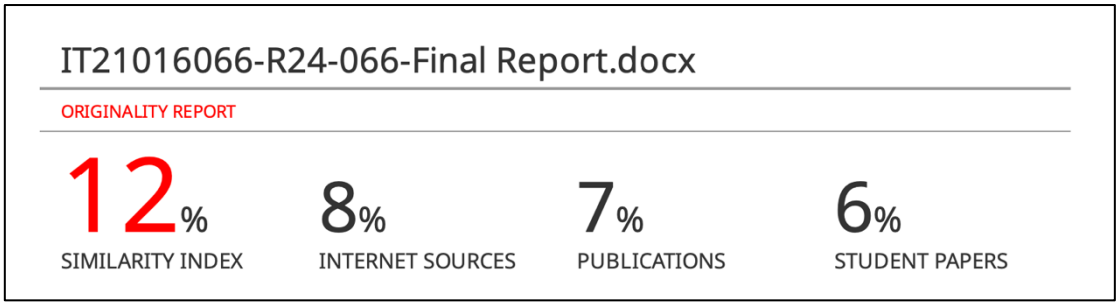
We will provide original demographics data after your PP1,till we provide demographics data you can continue your project using dummy demographics data.

Thanks & Regards

Gayan Ranaweera | Executive – Research & Development
LINEA AQUA (PRIVATE) LIMITED
Thanahenpitiya Estate, Giridara, Kapugoda, Sri Lanka
M: +94 71 0949587
W: www.masholdings.com/swimwear.html

LINEA AQUA MAS speedo

APPENDIX B: Turnitin Report



IT21016066-R24-066-Final Report.docx

ORIGINALITY REPORT

12%	8%	7%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS