Master Thesis

PREDICTING PRODUCT CHARACTERISTICS USING NEURAL NETWORKS : AN ANALYSIS OF MODELS EFFECTIVENESS FOR ZERO DEFECTS SCREENING STRATEGIES

Nikitha Umamahesh Ritty
Business Information Technology

Supervisors :
Dr.Faiza Bukhsh (EEMCS)
Dr.A.Abhishta(IEBIS)

June 2023

Faculty of Electrical Engineering,
Mathematics, and Computer Science,
University of Twente,
Enschede,
The Netherlands.

UNIVERSITY OF TWENTE.

PREDICTING PRODUCT CHARACTERISTICS USING NEURAL NETWORKS : AN ANALYSIS OF MODELS EFFECTIVENESS FOR ZERO DEFECTS SCREENING STRATEGIES

Master Thesis

Final Version

Enschede, June 2023

Author:

Nikitha Umamahesh Ritty (2711923)

Master of Business Information Technology

n.umamaheshritty@student.utwente.nl

University of Twente

Graduation Committee:

University of Twente

First Supervisor: Prof. Dr.F.A.Bukhsh

Second Supervisor: Prof. Dr.A.Abhishta

Acknowledgment

The submission of this thesis signifies the culmination of my master's studies at the University of Twente. This academic endeavor has been a transformative experience, providing me with invaluable life lessons. It all commenced in January 2023 when I commenced my internship at NXP.

I extend my sincere gratitude to my company supervisor, Fabrizio Finelli, and my university supervisors, Faiza and Abhishta, for their invaluable guidance and support throughout this research journey.

I am truly thankful to Fabrizio for offering me the opportunity to work on this research project during my internship at NXP. His expertise and unwavering support have been pivotal in shaping the trajectory of this study. His guidance and inspiration have significantly contributed to the quality and success of this research.

Faiza, I am truly grateful for your guidance, support, and valuable insights throughout this research. Your expertise in data science and Design Science Research Methodology played a pivotal role in shaping the direction of this thesis. I greatly appreciate your availability for discussions and feedback, and your mentorship has been invaluable. Your dedication to my progress and your insightful contributions have significantly contributed to the overall success of this thesis.

I would also like to express my appreciation to Abhishta for joining as my second supervisor in this research endeavor. Your expertise and insights have provided valuable perspectives to the project, enhancing its quality and depth.

To the entire team at NXP, I extend my gratitude for their support and cooperation throughout my internship. The opportunities provided by the company and the collaborative environment greatly enriched my learning experience and enhanced the quality of this research.

Finally, I am deeply grateful to my family, especially my mother, and friends for their unwavering support, encouragement, and understanding throughout this journey. Their belief in me and continuous motivation were invaluable in overcoming challenges and staying focused.

I would like to extend my sincere gratitude to all individuals mentioned earlier, as well as anyone else who has played a role in the completion of this thesis. Your contributions, support, and guidance have been invaluable in reaching this important milestone in my academic pursuit.

Contents

Abstract

As the significance of product quality and the demand for efficient and precise predictions increases, machine learning techniques have become a desirable solution for businesses. However, it can be difficult to determine the most suitable technique for a particular task. Prior research has highlighted the effectiveness of machine learning techniques in prediction tasks, prompting this research to focus on developing a predictive model using a neural network approach and exploring different approaches within the neural network. This research specifically investigates techniques related to data preprocessing and feature selection, with the aim of identifying effective approaches to optimize the performance of neural networks. The study evaluates various neural network architectures and associated criteria to determine the best configuration for accurate predictions. By systematically comparing these methodologies, it provides insights into their strengths and limitations, aiding decision-making in implementing neural networks for predicting product quality.

Recognizing the need for usability and accessibility, this research also proposes the development of a user interface for the predictive model. By creating a user-friendly interface, individuals with limited knowledge of machine learning can effortlessly harness the power of the model to make accurate predictions. This enhances the practicality and adoption of the predictive model in diverse business settings.

Furthermore, this paper sheds light on the challenges and limitations associated with employing neural networks in this specific domain. By recognizing and addressing these challenges, potential avenues for improvement and further research can be identified.

Keywords: *Product Characteristics Prediction, Neural Networks, Model evaluation, Efficiency, Reference Model*

1 Introduction

The semiconductor industry is in a constant state of change, driven by increasing demands for higher quality products. The automotive semiconductors market specifically pertains to the production and sales of semiconductor devices and components used in automobiles, which can include microcontrollers, sensors, and power management integrated circuits (ICs), among others. In recent years, there has been a notable increase in the adoption of advanced technologies for the mass production of cars, particularly those related to safety features such as collision avoidance and driver alerts, which use technologies like ADAS. As a result, there has been a surge in demand for semiconductor components in emerging markets, leading to expanded product offerings to meet this demand.

A recent report by Grand View Research indicates that the global automotive semiconductor market was worth USD 40.4 billion in 2020 and is projected to grow at a compound annual growth rate (CAGR) of 6.1% from 2021 to 2028 [37]. The report identifies factors such as the growing demand for advanced safety features, the rising popularity of electric vehicles, and the emergence of connected cars as key drivers of the market.

The global market for automotive semiconductors is characterized by a moderate level of competition, with several major players such as NXP Semiconductors NV, ON Semiconductors, and others. These companies are investing heavily in research and development to enhance and innovate their products, in response to the growing demands of the market. In this competitive environment, companies are required to make technology upgrades, significant capital investments, and develop scalable solutions to remain competitive.

The largest market for automotive semiconductors is the Asia Pacific region, followed by North America and Europe. This can be attributed to the existence of major automobile manufacturers in these regions, along with the rising demand for electric vehicles and the expanding use of advanced driver assistance systems.

The semiconductor industry has encountered significant challenges in recent times, primarily due to the COVID-19 pandemic, which has disrupted the production chain and led to shortages affecting various industries, including automotive. According to research conducted by the Korea Automotive Technology Institute, the shortage of automotive semiconductors has persisted from 2020, when the pandemic began, through 2022, with some companies still facing supply chain disruptions.

To meet the increasing demands, production capacity must be expanded, which can be a time-consuming and costly process that requires prioritizing the delivery of high-demand products. This can have a significant impact on safety and the capital invested in testing all circuit units to ensure defects are limited in any process.

The semiconductor industry is gradually adopting machine learning techniques to improve production processes, which has yielded positive results. One significant advantage is that it can greatly reduce the time and costs associated with manufacturing units. Additionally, by predicting the behavior of dies, machine learning can help to minimize the use of faulty units in further processes, which improves the overall quality of the final product. This optimization of performance can also reduce the need for expensive trial and error testing.

Machine learning can be used to analyze large amounts of data from different stages of the production process to identify patterns and correlations that may not be evident through manual analysis. By doing so, manufacturers can optimize their production processes and reduce defects, resulting in higher quality products and reduced costs.

It is crucial to minimize defects in automotive semiconductor devices, which can be achieved through zero-defect screening test strategies. Machine learning can be utilized in these strategies by examining data from previous processes and optimizing the screening process. By analyzing various production variables that impact defect rates, machine learning algorithms can identify areas for improvement and optimize the testing process to reduce defects in subsequent production processes.

1.1 Research Background

Over the years, the prediction of product characteristics has gained increasing importance in a variety of industries such as manufacturing, healthcare, and finance. Machine learning models, specifically those based on neural networks, have shown promising results in accurately forecasting product properties and behavior. However, there are several challenges that need to be addressed in this area such as the requirement of large amounts of data to train the models, the selection of appropriate features for the models, and the need for models that can adapt to changing product designs. Moreover, the complexity of products and their interrelated characteristics pose difficulties in accurately predicting product performance.

In order to overcome these obstacles, the research conducted in the area of product characteristic prediction has concentrated on different areas such as creating innovative machine learning algorithms capable of managing intricate data, discovering new sources of data to improve model accuracy, and exploring new techniques to choose and assess features. Moreover, there has been a particular focus on utilizing predictive models in particular industries and product categories, such as medical devices and consumer electronics.

To identify areas for future research and to address gaps in the current literature, it is essential to conduct a thorough review of the current state of knowledge in product characteristic prediction. This review should encompass a variety of topics, such as the types of predictive models employed, the data sources and features utilized in training these models, and the techniques employed to validate their accuracy.

1.2 Research Objectives

The semiconductor industry has recognized the need for technology that can identify units based on their behavior during the testing process. This technology can contribute to improved performance optimization and reduce the use of units with bad behavior in further processing. Manual solutions have been deemed inefficient, prompting the need for a more sustainable approach. One such approach is the development of a model that can predict product characteristics using a Machine Learning algorithm technique. Based on the findings of previous research [38], which primarily examined machine learning models, the researcher recognized the promising performance of the autoencoder as a solution to the research problem at hand. This observation led to identify a research gap that called for further exploration into neural network modeling techniques. Motivated by this knowledge gap, the present study is dedicated to a comprehensive investigation of neural network models, with the aim of advancing our understanding and application of these models within the specific context of the research problem. To make this approach accessible to non-developers, a self-developed user interface is also necessary. Thus, this research aims to achieve the following journal objective:

- Proposing a model for predicting product characteristics using neural network.

The objective within the semiconductor industry is as follows:

- Proposing a model for predicting product characteristics using neural network for screening method.

This study aims to contribute significantly to existing knowledge by incorporating insights from prior work and exploring the latest advancements in neural network modeling. Through in-depth examination of architectures and methodologies, the research seeks to uncover novel insights and approaches that enhance the effectiveness of addressing the research problem. By leveraging combined knowledge and techniques, the study strives to provide valuable contributions, leading to more effective solutions for the problem at hand.

To verify the potential benefits of utilizing a neural network model to accurately predict product characteristics, several steps must be taken. Firstly, a systematic literature review must be conducted to identify the relevant research journal publications that showcase the preliminary machine learning techniques used in this context. Secondly, the appropriate context for developing and implementing the model must be determined and validated. Then, a specific model must be formulated based on the developed reference architecture that is tailored to fit the context of predicting product characteristics. This concrete architecture must include the functional requirements for a functional model that conforms to the developed reference architecture. The model will then be tested on the dataset to validate the design decisions of the reference architecture.

1.3 Research Questions

In order to achieve the previously mentioned objective, it is necessary to first define the main research question. This main research question is crucial in establishing the necessary steps required to achieve the research objectives previously defined, and serves to highlight a set of sub-questions. The main research question of this research is formulated as:

- Which techniques can be employed to enhance the performance of neural networks for predicting anomalies in product characteristics and achieve zero defects screening strategies in the semiconductor industry?

This question is having sub-questions:

1. What are the most effective techniques for preprocessing product characteristic data for use in neural network training?
2. What are the limitations and challenges associated with the use of neural networks for predicting in product characteristics?
3. How can neural network architecture and design be optimized to improve the accuracy of predicting product characteristics for zero defects screening strategies?
4. How can the performance of the optimized neural network be validated and continuously used in real-time operation?

1.4 Research Scope

This research focuses on developing and validating a reference architecture that can aid in designing neural networks for predicting product characteristics by providing a flexible and adaptable architecture for the context. The goal is to develop a neural network architecture that can outperform traditional models and is capable of effectively handling new data. To achieve this, a sample population of products will be used to ensure the accuracy of the dependent variables in real-time conditions. It's important to note that this study only considers one version of the relevant architecture to be used in modeling. Building upon previous work [38] that examined traditional machine learning models, the researchers observed promising performance of the autoencoder and suggested further investigation into neural network modeling. Consequently, this study centers on exploring and advancing the neural network model.

1.5 Research Methodology

This section outlines the framework utilized in the knowledge domain, providing guidelines for performing the task at hand. Additionally, the research methodology that guides the progress and structure of the study is discussed. In order to facilitate comprehension, several research studies are used to establish a description of the reference architecture.

1.5.1 Reference architecture Design Framework

The utilization of reference architecture varies across different disciplines and is explained in different ways. Kaiming He et al. (2016)[6] defines reference architecture as a blueprint or conceptual framework that consists of best practices for operations and can be applied to multiple projects in analytical system development. It serves as a generic architecture with minimal architectural methods that serves as a foundation for designing a more specific architecture within the same context. Employing reference architecture can reduce modeling computation time and risk in projects within the same domain class, especially for organizations with limited resources. Reference architecture is typically created at a higher level of abstraction to enable its reuse and provide a reliable basis for future architecture designs. Conversely, an architecture designed to address a specific research objective in a particular context is regarded as a concrete architecture.

The architecture developed by Kaiming He et al. in their 2016 paper was designed based on the context, goals, and design. The context dimension of the architecture pertains the "How" and "Where" aspects of its development. The "How" categorizes reference architecture as classical or preliminary, depending on whether the algorithms have been validated experimentally. The goal dimension specifies the intended purpose of the reference architecture, while the design dimension outlines the level of detail, types of functions, degree of abstraction, and level of performance. The objective aspect of the reference architecture is defined by the goal dimension, which outlines the intended use of the architecture. On the other hand, the design dimension specifies several characteristics, such as the degree of abstraction, types of functions, level of performance, and level of detail.

1.5.2 Design science Research Methodology

Peffers et al. (2007) described design science research methodology as a problem-solving framework for conducting Information Systems research that aims to design and evaluate innovative artifacts, which can be used to solve practical problems and generate new knowledge. R.J.Wieringa (2014) provided an overview of Design Science research methodology as an approach to developing and evaluate Information System artifacts in specific contexts, involving cycles of designing, building and evaluating the artifacts. Also, stating the goal of the design science methodology as to create new knowledge about the design and use of information systems, achieved through the construction of unprecedented artifacts and their assessment in experimental environments. Design science projects generally involve three main tasks: problem investigation, treatment design, and treatment validation, which are carried out in a cycle of design and investigation, as depicted in Figure 1.

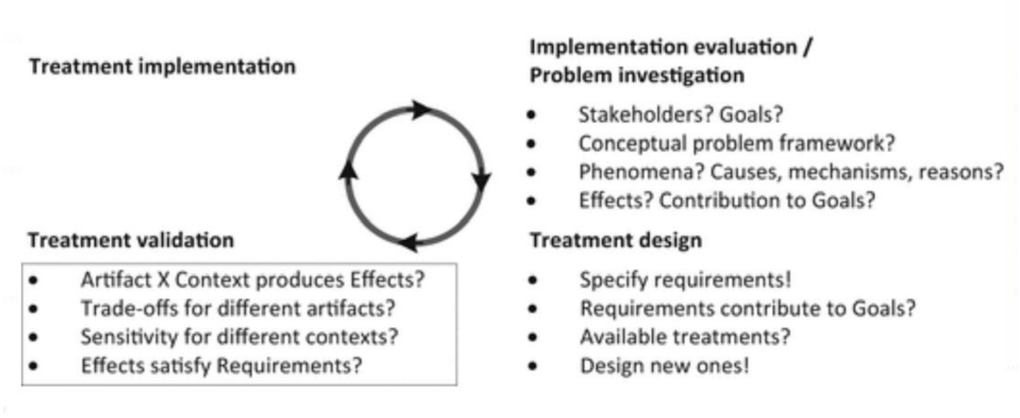

Treatment implementation
Implementation evaluation /
Problem investigation
Stakeholders? Goals?
Conceptual problem framework?
Phenomena? Causes, mechanisms, reasons?
Effects? Contribution to Goals?
Treatment validation
Artifact X Context produces Effects?
Trade-offs for different artifacts?
Sensitivity for different contexts?
Effects satisfy Requirements?
Treatment design
Specify requirements!
Requirements contribute to Goals?
Available treatments?
Design new ones!

Figure 1 Design science engineering cycle (R.J.Wieringa, 2014)

R.J.Wieringa (2014) emphasizes the importance of research method as it enables researchers to provide comprehensive and structured answers to research questions in a rigorous and systematic way. The Single-Case Mechanism Experiment is a research method that can be used in design science to evaluate the credibility of an architecture implementation. It involves testing the validity of a design artifact in a specific real-world context. This mechanism is tested on a single object of study with a known architecture. The validation model involves an artifact prototype and a context model. The artifact prototype works with the problem context model to evaluate the validity of the interaction between the artifact model and the context model. This helps ensure that the implemented artifact interacts properly with the real-world problem context.

The validation, artifact, and context models must be evaluated in specific scenarios within the case study to determine their validity. The researcher conducting the validation should select an application scenario to test the context model and determine the necessary measured variables and scales required to assess the performance of the validation model. These can be predefined as part of the artifact's design, but in this particular study, additional constructs and indicators must be included in the proposed artifact to quantify the measurement.

1.6 Thesis Structure

The first chapter of this document introduces the research. The following chapters have specific focuses: Chapter two outlines the research methodology, including the research scope and methods utilized to gather information used in the document. Chapter three provides background knowledge relevant to the research. Chapter four discusses the pre-processing of data that needs to be done before continuing. Chapter five describes a user interface that makes it easy to use the designed model. Chapter six presents the model architecture and results obtained from the designed model of this research. The final chapter concludes the research and reviews the research questions and their corresponding conclusions.

2 State of Art Prediction Product Characteristics using Neural Network Reference Architecture

As stated previously, the current state-of-the-art reference architecture for Neural Networks will be derived from scientific journal articles. To achieve this and obtain the necessary components for the general reference architecture of Neural network, a systematic literature review(SLR) will be conducted in this chapter. The following sections will describe the methodologies used of this SLR, along with the analysis of the results and the final reference architecture for the Neural network.

2.1 Methodology

In order to perform a systematic literature review, few steps were followed as mentioned in Kitchenham et al. [1]. These steps included defining research questions, assessing studies collected in a critical manner, and utilizing search strategies, inclusion and exclusion criteria, and quality assessments for the studies were included where the process is categorized into three main phase: Planning, Selection and result analysis. Table 1 outlines the detailed activities involved in the phases, which will be further elaborated on in the subsequent sections with a more in-depth explanation of the particular approach used in conducting the systematic literature review, which accentuates its thoroughness.

Planning	
1	Defining main Research Question and sub-questions
2	Select scientific databases
3	Formulate search query based on main research question
4	Define inclusion and exclusion criteria
Selection	
5	Execution of search query in each scientific database
6	Articles selection for each query resulted by inclusion criteria
7	Remove of duplicate studies across scientific database
8	Exclusion of irrelevant articles based on abstract and introduction assessment
Result Analysis	
10	Data Extraction
11	Amalgamation of the extracted data
12	Integrated results for the research objective

Table 1 Systematic Literature Review Activities

2.2 Planning

This section is dedicated to outlining the goals and methodology of the review. Firstly, it involves the identification of research questions, followed by the selection of appropriate scientific databases for conducting the search queries, along with the defining criteria used for include and exclude search results.

2.2.1 Research Question

One Main research question was formulated and sub-questions for the critical analysis for research with respective prioritization method and its evaluation.

RQ1. Which techniques can be employed to enhance the performance of neural networks for predicting anomalies in product characteristics and achieve zero defects screening strategies in the semiconductor industry?

SQ1. What are the most effective techniques for preprocessing product characteristic data for use in neural network training?

SQ2. What are the limitations and challenges associated with the use of neural networks for predicting in product characteristics?

2.2.2 Approach

The research domain is being thoroughly explored and comprehended by expanding the scope as much as feasible. This has including the use of machine learning as a model in all fields of service. In order to address the research inquires, certain criteria have been established as inclusion and exclusion criteria. Once the criteria have been established, a query will be created using these criteria, which will be executed on our chosen database. This query will help to classify articles or journals based on certain inclusion and exclusion criteria, which will refine our search towards our domain.

Articles or journals are chosen based on their abstract and data content. This allows for a better understanding of the topic and enables researchers to extract relevant information to answer the main research question as well as sub-questions. This information can then be utilized to develop solutions to the research questions.

2.2.3 Scientific Database & Search Query

Based on research question, some of digital libraries was selected to broaden the understanding in the selected subject by the use of their publications mainly that are journals and conference proceedings. This libraries were selected as a well-known library as mentioned in paper[2] being one of the top thorough and accessible databases which are most trusted academic resources databases.

In this research, IEEE, Scopus, ScienceDirect and Springer are used. The main keywords that was used in the search strings was : “Machine Learning” and ‘’Product Characteristics prediction” as there were the suitable combination of keywords to refine the research papers.

“(TITLE-ABS-KEY (product AND characteristics AND prediction) AND KEY (machine AND learning))’’, the mentioned query was used to obtain all the papers pertaining to the topic without omitting any of them. As a result of this search, 488 papers were found in Scopus.

To restrict the paper to the selected topic, some of the following constraints were imposed in defining the research scope. To begin with employing filters to refine the document type to either conference papers or research papers, focusing on the subject area of Computer Science. The second limitation was confining the papers to a period of 10 years, starting from 2013 to 2023.

Another limitation imposed on the research scope was the language of the papers, whereby only those written in English were considered and all others were excluded. Lastly, search was limited to open access papers as this would make it more accessible and user-friendly for everyone to access the papers. As a consequence of this search, 133 relevant documents were obtained, with any irrelevant documents excluded.

Similar search queries were used in different libraries and obtained different search outcomes from each library. This required trying out different combinations of keywords to achieve the desired search outcomes. Some of the libraries resulting in higher number documents the keywords filters were used to pertain correct documents, where the neural networks was used to narrow down the documents.

2.3 Selection

At first, the research scope was confined to the topic of interest. Once all the documents were collected from each library, the duplicate papers were eliminated, and the inclusion criteria (ICs) and exclusion criteria (ECs) were then applied. The inclusion criteria utilized to select the papers, are as follows:

- IC1 The papers that proposed an approach or implementation to solve problems related to the research question were included.
- IC2 The paper is a journal, conference or research articles.
- IC3 The paper is in English
- IC4 The paper is open access.

The exclusion criteria applied on the papers are:

- EC1 Research studies that do not pertain to the main research question, based on the title, abstract, and content.
- EC2 Articles that too short to support the research
- EC3 Duplicates articles and paper not peer-reviewed.

This criteria's were applied by reading the abstracts and introduction of the collected articles from the libraries. This articles were categorized and by the end of the categorization among the collected 131 articles, there were 34 articles selected and entire process is illustrated in the Figure 2.

To evaluate the quality of the collected articles that will be used to answer the formulated research question, a critical appraisal on this articles was conducted. For this purpose, a set of assessment questions related to the research question were formulated, and were applied to each article. The purpose of this assessment was to provide feedback on the quality of the articles.

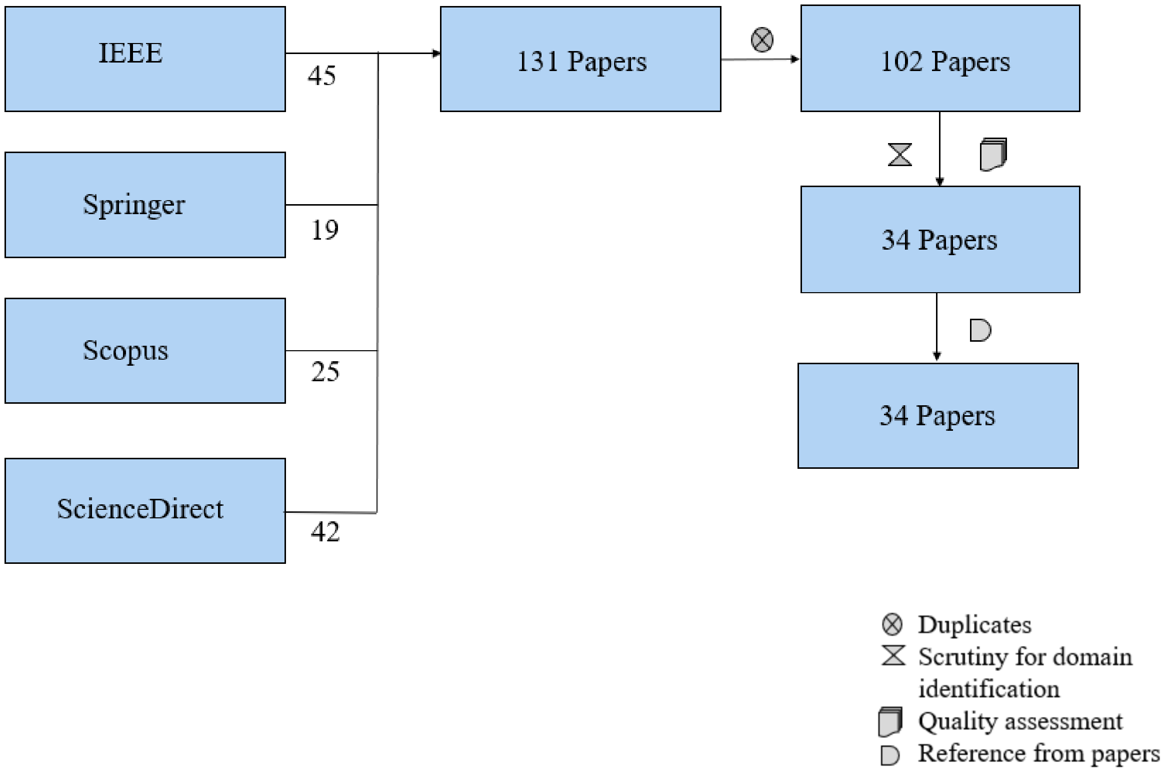


Figure 2 Scientific Database Articles Extraction

Based on research question,

- Does the paper discuss various neural network techniques?
- Does the paper strongly assert the effectiveness of the proposed neural network?
- Does the paper elaborate on the differences in neural network architecture layouts and how these differences impact the results?
- Is the proposed evaluation method discuss about model being sufficient for assessing the neural network model?
- Does the paper provide a comprehensive comparison of models that affect prediction results?
- Does the paper address the limitations of the proposed model?

These questions being important techniques that are applied on the collected articles to perform analysis of quality for the articles. Based on this techniques, articles were carefully filtered among all the literature papers that are found from the libraries. If there were articles that could answer some of the questions instead of every quality assessment questions, the articles were selected. But when the articles were having inadequate answer and not support the decision of the proposed techniques it was excluded.

With the application of the inclusion and exclusion criteria and following with the selection of articles, relevant information that are important in addressing towards the formulated research question is to be collected. As a result of the varied objectives of this articles, different outcome can be discerned from this literature articles. The outcome of each study is being identified to evaluate the efficacy of their solution and to appraise the quality of the research.

Out of the 133 papers that were selected, the research methodology, outcomes, and observations were identified and reviewed to gather relevant information to address the sub-questions and validate the method's application. These findings will aid in addressing the research question and evaluating the proposed solution.

2.4 Data Extraction

The aim of article selection is to collect pertinent information that addresses the research questions. The extracted content will be utilized to formulate a reference architecture through a synthesis process, which will be discussed in the next section. A form is presented in the following Table 2 to evaluate the relevance of the selected papers in constructing the reference neural network model. The research method utilized in this review is observation and experimentation to determine how the findings from relevant studies are formulated. The data collection process is explained in detail in the Table 2. The extracted data is listed in appendices

No.	Data extraction		Depiction
1	Citations		Authors and publication year
2	Research Objective		Proposed approach to problem and relevance towards Neural Network prediction
3	Research Methodology	Experiment	Technique implementation and measure of the influence on their case
		Observation	Judgement of the outcome
4	Research Result	Academic	Principles, advantages and disadvantages of the model
		Architecture	Proposal of design for the study incorporating design proposed in the literature
		Theoretical Framework	Design decision using model representations to supplement the theory
		Working Implementation	Process of implementing solution explanation
5	Study outcomes	Drivers	Objectives for the involvement of initiative of the study
		Framework components	The determination of structure and pattern of the model architecture
		Automation	Technology employed for resulting outcome

Table 2 Quality Assessment and Data Extraction Form

2.5 Mapping Literature with Research questions

Following the selection of articles, the relevant information is gathered to address the research questions. This information will be utilized to develop a machine learning model employing a neural network, which will be further discussed in a later section. The articles selected for this study, along with their research methods is analyzed and the information that is referenced for designing is discussed in this section. First, the motivation for identifying them in this review to examine how the studies in this field are generating their outcomes and drive reference architecture from the literature. Secondly, the analyzed information from the articles is evaluated to determine its potential applicability from references in addressing the first research sub-question (section 2.5.2) and formulating a structured architecture that can serve as a guide for treatment design. Thirdly, analysis providing a comprehensive understanding of the challenges associated with utilizing neural networks for product characteristic prediction, which can inform future research and development in this field, effectively addressing the second research sub-question (section 2.5.4).

2.5.1 Motivation

To explain the rationale behind obtaining a reference architecture from literature that impacts and limits the design of the neural network model, several key concepts need to be addressed. These concepts encompass multiple aspects that are crucial in expanding the discussion. The initial concept is the driver, which is an internal or external condition that inspires the development of a more productive and proficient model by establishing its goals and executing essential modifications. The goal is a statement of purpose or intended outcome to understand the importance and benefits of using the literature support. The following Table 3 presents a summary of the extracted data, specifically the identified drivers and goals, along with the references from which they were derived.

No.	Driver	Goal
1	Adaptability	Better understanding of the current state-of-the-art [3][5][7][14][17][19][25][27][30][34]
2	Leveraging existing knowledge	Access for expertise and knowledge for the field [4][5][11][13][16][32][33][30]
		Common themes & principles[8][9][12][18][22][24][31]
3	Reduce flaws	Guidance in development [7][20][23][26][28][31][34][36]
		Improve performance[15][21][35]
4	Reliability	Trustworthy model [14][19][24]
		Increase efficiency[10][24]

Table 3 Driver, Goals of prediction product characteristics adoption

The information gathered from the literature studies indicates that the reason for utilizing models is to enable the creation of a comprehensive understanding of the currents state-of-the-art techniques, algorithms, and approaches used in the field of neural networks. The novel techniques introduced in this study for structuring the model improve its generalizability by examining studies across various domains, identifying common themes and principles that can be applied more broadly, and beyond specific applications.

The utilization of literature support in developing a neural network model provides guidance for creating an accurate and efficient model, identifies potential limitations, and avoids common mistakes made during the modeling process. This approach ensures that the model is based on validated and reliable information, resulting in a robust and trustworthy model. The established techniques and best practices from previous studies inform the design of the new modeling method, increasing the likelihood of success. Additionally, the literature support provides a foundation for future research in modeling, allowing for the advancement of the field and significant contributions to its development.

2.5.2 Preprocessing Techniques

Analyzing the most effective strategies for answering the research question from a range of methods discussed in literature for problem-solving. The approaches used in the 34 papers was dependent on the specific problem context. The techniques applied were chosen based on the available data to create a model that would yield the best results.

Preprocessing involves several techniques aimed at optimizing the preparation of product characteristics data. Within the literature, several effective techniques were identified for effectively preprocessing such data. One of these effective techniques, discussed in [7], involves dealing with empty data. Instead of imputing missing values, they suggest to remove the empty data. This approach was considered beneficial as it allowed for elimination of incomplete or unreliable data, ensuring that the model was trained on high-quality and meaningful data points. Similarly, another reference[15] also proposed similar techniques in the handling of data null values, which involves removing and scaling them within a specific range. Removing null values ensured that they did not introduce bias or affect the analysis. Scaling the data within a range, such as 0 to 1, was advantageous in machine learning processes, as it standardized the values and prevented certain features with larger magnitude from dominating their learning process. Scaling the data aided in achieving better convergence and improving the overall stability of the model. By removing these rows with missing values and scaling values, the model was able to focus on learning from complete and relevant information, potentially improving its performance and accuracy. These techniques for handling empty and null values are referenced from the literatures and have shown effectiveness in dealing with product characteristics data in their domain.

As in reference[29], they discussed techniques such as Principal Component Analysis, Correlation-based Feature Selection, and ReliefF, which were used in combination with traditional methods and neural network for feature selection. It is worth noting that even after applying feature selection, the selected features has uncontrolled influences on the outcome. The traditional method exhibited better performance, while the neural network model showed a worst-case evaluation metric. Despite the effectiveness of these techniques in handling product characteristics data, it is important to acknowledge the possibility of uncontrolled features that may exist even after feature selection.

In the realm of feature selection and dimensionality reduction, the employment of machine learning algorithms techniques, specifically random forest and AdaBoost has been proven to be highly effective as demonstrated in the experiments conducted by reference[7]. The given dataset in their research comprised continuous data with notable fluctuations, making identification of relevant features paramount. Through evaluation among various techniques, random forest and AdaBoost were recognized as powerful algorithms for feature selection, outperforming others methods and producing the highest performance. The evaluation of each feature's contribution was carried out by measuring how much it reduced impurity or uncertainty in the predictions made by those features. This information was subsequently utilized in ranking the features based on their importance and to identify the top features within the dataset. Through the application of these techniques, the researchers successfully identified and extracted the crucial features from the dataset, leveraging the capabilities offered by the methods. This process resulted in dimensionality reduction and more focused set of features for further analysis. The use of random forest and AdaBoost algorithms, coupled with careful evaluation of feature contributions, enabled the researchers to optimize their model by utilizing only the most informative features, ultimately improving the accuracy and interpretability of their results. The findings from reference [17] provides evidence of the effectiveness of dimensionality reduction. In reference [21], the random forest algorithm was selected to determine the most important factors in ranking the importance of test metrics. By examining the feature importance's provided by the random forest model, it was possible to identify the subset of features that have the most significant impact on the target variable. By focusing only on the most relevant features and disregarding less influential ones, the dataset's dimensionality is reduced. The strength of the random forest algorithm, as mentioned in reference [24], lied in its ability to handle high-dimensional data with a large number of features. It effectively handled datasets containing correlated, noisy, or irrelevant features. This is achieved by utilizing multiple decision trees that collaborate to reduce individual tree biases and variances. Random forests serves as valuable technique for dimensionality reduction and feature selection, enabling researchers to gain insights into the most influential features and effectively manage complex datasets. It offers a robust framework for feature selection, leading to improved accuracy and interpretability in different domains.

Data imbalance referred to an uneven distribution of classes within a dataset, which creates challenges during model training. Some literature offered recommendations for addressing this issue by utilizing techniques specifically designed to overcome the difficulties posed by imbalanced data. In study[26], the researchers conducted an experiment to address data imbalance in their dataset by employing random under-sampling, which involved randomly selecting a subset of samples from the majority class. Additionally, they utilized an alternative techniques called Self-Organizing Map (SOM) clustering to group similar samples based on their feature vectors, which helped reduce the impact of data imbalance. This approach proved advantageous as it allowed for oversampling or under sampling techniques to address data imbalance without significant loss of information. However, the study encountered limitations in applying these techniques to high-dimensional data, which made the approach less feasible and impractical. As a result, the researchers focused on using low-dimensional data in their study to overcome these limitations.

Another data imbalance technique mentioned in reference[7] is Synthetic Minority Over-sampling Technique (SMOTE). In contrast to random under-sampling, SMOTE generated synthetic samples from the minority class to augment its representation in the dataset. One notable advantage of SMOTE was its ability in creating a substantial amount of synthetic samples, even when the original minority class samples were limited. By augmenting the representation of the minority class, SMOTE mitigated the issue of data imbalance and enhanced the models performance. Through their experimentation, they obtained evidence that the inclusion of SMOTE led to superior model performance compared to the original dataset. This was particularly prominent in scenarios involving high-dimensional data. The choice of data imbalance techniques depends on the factors such as the dataset characteristics and the dimensionality of the feature vectors.

It's important to note that the selection and applicability of the techniques differ depending on the specific characteristics of the given dataset and the requirements of the problem discussed in the references. To validate the effectiveness of these techniques, experiments can be conducted using the given dataset, involving the designing and training a model using a reference architecture as benchmark. Throughout the experimentation phase, the model can be adjusted accordingly to best suit the context of the problem.

2.5.3 Reference Architecture

The reference architecture is derived from literature review, showcasing their model design and its efficiency in addressing various domain-specific problems. These studies validate the architecture by employing different performance metrics and demonstrate its effectiveness in solving their respective problems. One particular reference model highlighted in papers [7][14] and [18] emphasizes the use of feed-forward propagation, showcasing promising results with a single hidden layer. There are many models utilized in machine learning[34], and their effectiveness varies over time due to technological advancements and domain requirements. In the past decade, deep learning and neural networks[22] have gained popularity and are currently the most commonly used models in various domains. Since 2013, their usage has increased rapidly, surpassing other models in popularity. However, some models remain consistent in their usage and have not experienced significant growth over time. Another study [3][20], explores various machine learning approaches and identifies neural networks as the best choice, particularly with 10 and 20 hidden layers, outperforming other models in their research. To develop a neural network model that is efficient and effective, the architecture can be useful in identifying the essential resources and design patterns. It can also provide a clear understanding of the various layers and components of the neural network, as well as the training and validation processes involved. This, in turn, aid in the creation of a high-performing model that satisfies specific requirements and research objectives. Additionally, incorporating the dropout function during training helps prevent overfitting, as mentioned in [9][33]. The LSTM training can be positive training depending on the optimizer, [33] suggesting to use the RMSprop optimizer with a batch size of 100. Furthermore, [34] recommends utilizing binary cross-entropy as the loss function in conjunction with sigmoid activation. These choices result in improved optimization effects, particularly in classification datasets. This not only saves time but also reduces computational efforts by providing a starting point rooted on past successful implementations and empirical evidence. Additionally, setting the dropout rate to 0.5 and a batch size of 128, as suggested in [23][34], further enhances the performance preventing over-fitting[35].

Respective to the epochs [30], there is no specific limit, and it can vary depending on the model. Generally, the range of epochs can be set between 500 and 2000. Going beyond 2000 may result in longer training time and potentially lead to overfitting of the model. When it comes to data splitting for training and testing, a commonly recommended approach is to allocate 2/3 of the data for training and 1/3 for testing. This split ensures a balanced representation of data for model training and evaluation and using confusion matrix for the validation. In terms of determining the number of nodes in the hidden layer, there are various approaches [31][21][25]. One such approach suggests using a formula-based calculation, specifically the formula $m = (\sqrt{n} + l) + \alpha$. Here, m represents the number of hidden layer nodes, n is the number of input nodes, l is the number of output nodes, and α is a constant between 1 and 10.By using this formula, the optimal number of nodes can be determined through a trial and error process.

According to [32], the neural network architecture shows resemblance with few deep learning methods that also employ a flexible two-branch neural network design. However, there are some distinctions in terms of the specific learning rate, solver, and hidden layer sizes used in their approach. The architecture in [32] utilizes a single loss function, which limits its applicability to multi-task learning, and relies on the scikit-learn library for implementation. The reference architecture discussed in the paper outlines various approaches to neural network modeling, some of which exhibit similarities in their design. However, it is important to note that there are variations in the validation techniques employed. This indicates that factors such as sample size and model configuration can have a significant impact on the model's performance. Analyzing the sample size and model configuration is crucial for understanding how these factors influence the model's effectiveness. By considering these aspects, researchers can make informed decisions regarding the architecture, validation methods, and other design choices to ensure the optimal performance of their neural network models.

As suggested in the paper [15], it is advised to conduct hyperparameter tuning to determine the optimal parameter configurations for the architecture. The choice of parameters can vary depending on the specific modeling task, and fine-tuning them can significantly impact the performance outcomes. By carefully adjusting the parameters, researchers can explore different paths and configurations to achieve the best possible performance for their neural network model. This recognition of the importance of parameter tuning in the modeling process acknowledges that different parameter settings can lead to variations in the model's performance.

By synthesizing and integrating the relevant articles, the extracted contents play a pivotal role in formulating the reference architecture and guiding the subsequent design of the neural network model. This ensures that the model is constructed based on a robust foundation of literature. Reference architecture is proven to be effective in their context, capturing the best practices and principles in modeling design ensuring the designed model aligns with the established standards and incorporates the recommended approaches for achieving optimal performance avoiding reinventing the wheel by providing roadmap for model design.

The reference architecture serves as a valuable foundation for designing and constructing a neural network model. It provides a standardized framework and guideline that streamlines the modeling process, ensuring that the research questions and objectives are effectively addressed. The design of the neural network model is guided by the reference architecture, incorporating insights and lessons learned from it. By leveraging the foundation established by the reference architecture, researchers can make informed decisions and successfully navigate their modeling endeavors. Additionally, it aids in minimizing the possibility of errors and inconsistencies that could result from non-standard design and implementation practices by building upon existing knowledge and adapt the reference architecture to suit with the specific research context, can enhance the modeling design by incorporating necessary modifications and additions.

2.5.4 Challenges and Limitations on Neural Network modeling

Just like any other field, there are constraints associated with utilizing neural networks to predict product characteristics. When it comes to predictive analysis using machine learning models, two crucial criteria for determining the suitability of the model are "flexibility" and "interpretability." Some of the models relationship of flexibility and interpretability is being graphed in the Figure 3.

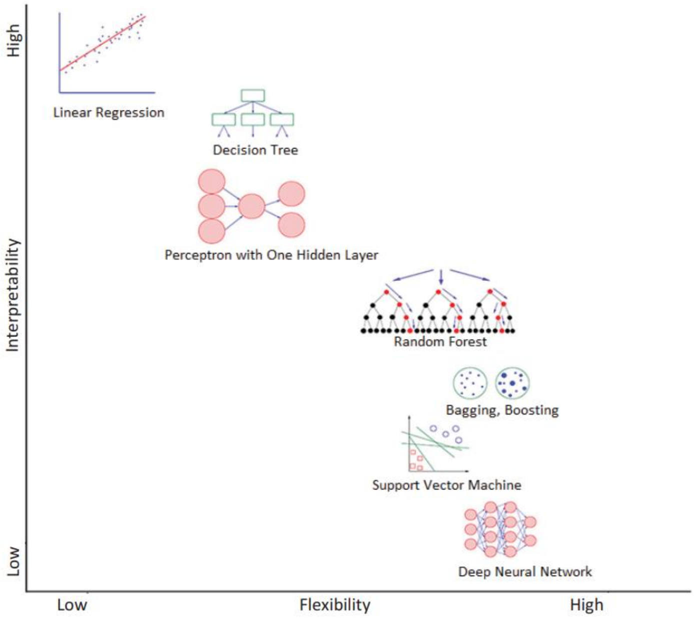


Figure 3 Relationship of model flexibility and interpretability for ML model ([19])

'Flexibility' is referred to the ability of model being able to adapt and generalize to new and unseen data. When the model is flexible it can capture complex relationships of input and output variables handing outliers. 'Interpretability' is referred to the ability that the model can be understood and explain how the model works and makes its prediction. Interpretability is important as they help to build trust in the model enabling better decision-making and facilitates the identification of errors and biases.

High flexibility leads to low interpretability of model, as it fits training data very closely but leads to overfitting and difficult to understand the result that is derived by the model making the predictions. This flexible models rely on complex interaction between features to identify important features for prediction making it difficult. This can limit the ability of domain experts to use the model to make decisions or draw insights from the data. Therefore, it is important to strike a balance between model flexibility and interpretability to ensure that the model is both accurate and understandable. Various techniques have been developed to improve the interpretability of complex models, such as feature importance analysis, model visualization, and explanation generation.

To achieve precise predictions in neural networks, it is essential to have abundant high-quality training data. Inaccurate predictions may result from incomplete or noisy data, which could require expanding the sample size for research purposes. Neural networks may struggle to generalize to new variations that were not included in the training data. Comparing results between two models, it was discovered that features from the z-axis had a weak correlation with quality characteristics and learning effects. When only z-axis features were used to train the model, the prediction results were very poor [19].

Neural networks demand expertise in data science and machine learning, which can make it challenging for non-experts to develop and implement them. The model's configuration takes into account certain conditions and may be affected by the restrictions imposed by the computer's configuration. Increasing the number of layers in the model or extending the training time can lead to some degree of improvement.

Training and running neural networks can be computationally intensive, especially for large datasets. They tend to produce more accurate results with larger numbers of neurons, but this comes at the cost of longer computation time. However, using smaller numbers of neurons can save time, but this may lead to less accurate results. Selecting the optimal neural network architecture and setup can be difficult, as various models may be better suited for different types of problems.

2.6 Summary

- State-of-the-art of neural network modelling from literature is derived, and a research question is formulated is formulated with two sub-questions:
  - SQ1. What are the most effective techniques for preprocessing product characteristic data for use in neural network training?
  - SQ2. What are the limitations and challenges associated with the use of neural networks for predicting in product characteristics?
- The primary scientific database is chosen, and literature from the database is selected using relevant keywords. A systematic literature review is then conducted, utilizing inclusion and exclusion criteria to filter the data, and relevant information is extracted.
- Techniques such as data removal, scaling, feature selection, and data augmentation were discovered to be effective for preprocessing the data.
- Various feature selection techniques discovered through research, such as Random Forest classifier, AdaBoost, Principal Component Analysis, Correlation-based Feature Selection, and ReliefF, have proven to be effective approaches for preprocessing product characteristic data before employing it in neural network training.
- Achieving accurate predictions in product characteristics using neural networks poses challenges and limitations. These include the need to balance the trade-off between model flexibility and interpretability. It requires expertise in machine learning and data science to effectively train the models with high-quality configurations.

3 Background Knowledge

In order to comprehend the focus of this research, it is necessary to present crucial details that are significant to the study. These components play a crucial role in gaining a better understanding of the research topic. Moreover, since the research objective is to neural network for predicting product characteristics, it is essential to have a comprehensive understanding of the product and the modelling method that is in details in the research question. The details is explained in subsequent section.

3.1 Preliminary Concepts in Semiconductors

3.1.1 Wafer

The foundation for semiconductor technology is the wafer, a thin, flat piece of silicon that acts as a substrate for creating electronic components such as integrated circuits (ICs). The ICs are the building blocks of electronic devices and consist of small components such as transistors and diodes that are etched onto the wafer's surface. Subsequently, any excess or unnecessary material is removed, which results in a wafer that contains multiple copies of the same IC design. These wafers are usually circular in shape and can range in size from a few millimeters to over 300 millimeters (12 inches) in diameter. To create the desired circuitry, the wafer undergoes a thorough polishing and processing procedure, which involves the deposition of multiple layers of materials onto the wafer, followed by the use of lithography to pattern the materials. Once the components are created, the wafer is diced into individual pieces known as dies, each containing one or more ICs. Before they are used in devices, the dies are tested to ensure they work correctly.

3.1.2 Dies

Dies is a small piece of semiconductor material, typically square or rectangular in shape, that contains the electronic circuitry for a single IC fabricated using lithographic techniques. They are created by separating the individual copies of the IC design that are present on the semiconductor wafer. The die contains electrical connections necessary for the circuit to function and is usually encapsulated in a protective package to ensure reliability and durability.

3.1.3 Wafer Testing

Wafer testing is an essential step in the manufacturing process, allowing manufacturers to detect and eliminate any defective dies before they are packed and shipped. The testing process involves subjecting the dies to series of tests, to ensure they meet the specifications for the intended application.

The wafer testing process typically begins with a wafer prober, which is a machine that electrically probes each die on the wafer to measure its electrical properties. The wafer prober uses special test equipment and software to apply a voltage to the die and measure the current flowing through it. This process is repeated for each die on the wafer, and the results are recorded for analysis.

The results of the wafer testing are recorded in a file called a wafer map, which categorizes the passing and non-passing dies by using bins to define good and bad dies. The wafer map is sent to the die attachment process, which selects only the passing circuits for further processing. Once the testing is complete, the results are analyzed to identify any defects or faults in the ICs. If defects are found, the wafer may be discarded. If the ICs pass the testing process, they are sent for packaging and the packaged dies undergo further testing during the final test phase as in Figure 4. Typically, the same or similar test patterns used during the wafer test phase are used during this final test phase.

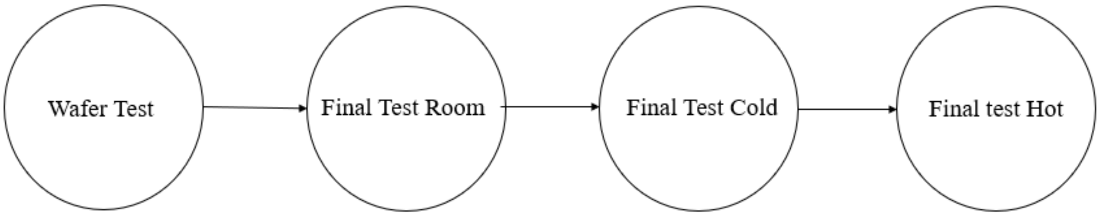


Figure 4 Wafer Testing Process

Although this process can be time-consuming and expensive, it is crucial for maintaining the quality and dependability of the end products, as it helps identify any issues in the manufacturing process that could lead to the production of faulty ICs. Advanced and automated methods, such as machine learning algorithms, are increasingly being used in wafer testing analysis to improve the efficiency and accuracy of the testing process.

3.1.4 Final Test Room

Final Test Room is a method of testing dies in testing process. This testing is processed in room temperature and is used to check the functionality of the circuitry on the dies. This testing phase is important step in the process as this ensures that the circuitry works as expected under normal operating conditions. The results of this final room testing are used to determine whether the dies are ready for further processing or if they need to be rejected due to defects or other issues. By performing final room testing, it ensures that the products meet the high standards of quality and reliability that is required.

3.1.5 Final Test Cold

Final Test Cold testing is similar to final test room but it is performed at low temperature to check the functionality of the circuitry on the dies under cold conditions. During final test cold, the dies are placed in a cold environment, typically a temperature controlled chamber and the circuitry of the dies is tested to ensure that it works as expected under these conditions. The results of final test cold can help to identify any potential issues with the circuitry in the dies under cold conditions, which can then be addressed before the devices are moved for next testing phase and ensure that they perform as expected under a wide range of operating conditions.

3.1.6 Final Test Hot

Final test hot is a process carried after the Final test Cold during the testing process, in which the dies is subjected to high temperatures to check the functionality of the circuitry under extreme heat conditions.

This type of testing is typically done at temperatures above the normal operating range of the device, and it is used to ensure that the device will perform reliably under the most extreme conditions that it may encounter during use.

During final test hot testing, the dies is placed in a chamber that is heated to a high temperature, typically around 125 to 175 degrees Celsius. The circuitry on the dies is then tested to ensure that it functions correctly under these high-temperature conditions. If any defects or malfunctions are detected, the dies may be rejected or sent back for further processing to correct the issues. Final test hot is important for ensuring the quality and reliability of semiconductor devices that may be subjected to high temperatures during use, such as automotive or industrial applications.

3.1.7 Screen Strategy for detection of Outliers

As most screening strategies is focused in identifying the outliers it is important to use this screening methods to test on good reproducible units producing higher quality and more efficient integrated circuits (ICs). This helps company to optimize testing processes in production, improve product quality and reliability, and manage supply and demand for semiconductors.

NXP semiconductors have split the defect screening in three main categories to follow with the safety and reliability standards [38]:

1. Parametric Screening- Outlier Detection:

This screening values are taken as input for the screening methods, such as correlation testing, static and dynamic limits screening. The goal of this screening is to find the outliers in the dataset. The outlier behavior is identified by the defects present in IC and the IC is assigned as failed BIN.

2. BIN-Level -Bad neighborhood rejection:

In this screening, the BIN results are used to reject materials in the bad regions neighborhood. These screening methods are effective for clustering of defects. If a good dies is found in a neighborhood of bad devices, there is a probability that the good dies is also defective. This could occur for mainly two reasons:

- A defect is present but that was not identified in the test program because they in non-tested node.
- A defect is present, but they are not active yet known as latent defect

3. BIN-statistical screening:

This screening is similar to the Bin level screening, but this is applied on wafer and batch level. When applying certain screening methods, the design must be enabled to do. Taking voltage stress as one of the example, when a voltage is not set to a specific value on certain nodes in the design due to voltage regulators could not be by-passed. This will impact the usefulness of voltage stress as screening methods and by that it endangers the quality level of the products.

3.2 Machine Learning

Machine learning is branch of artificial intelligence that focuses on creating algorithms and models that is capable of learning and improve from the experience without being explicitly programmed. Its growth has been fueled by big data availability and advances in computing power. With diverse applications in finance, marketing, and other fields, machine learning algorithms and fundamental concepts will be explored in this section, including the algorithm utilized in this research.

3.2.1 Types of Machine Learning Algorithms

Machine learning has three main types of algorithms, Supervised, Unsupervised and Reinforcement learning. Supervised learning is an approach where the algorithm learns from labelled data to make predictions or decisions. Within supervised learning, two main types of algorithms exist: Classification and Regression. Classification is the process of assigning input points to predefined categories. Algorithms like decision trees, logistic regression, and k-nearest neighbors are commonly used for classification. Logistic regression, a statistical technique, predicts categorical outcomes based on independent variables. Classification is widely applied in fields like sentiment analysis, spam filtering, and fraud detection. However, the choice of algorithm depends on data characteristics and the specific problem, as each algorithm has its strengths and limitations. Careful consideration is crucial for selecting the most suitable algorithm. Regression algorithms are used to predict continuous numerical values based on input data. They analyze the relationship between dependent and independent variables to find the best-fit line or curve that describes the data. The goal is to establish a mathematical model that can make predictions for future outcomes using past observations. Regression algorithms include linear regression, logistics regression, and polynomial regression, with linear regression being a commonly used method to find the best-fit line representing the relationship between two variables.

3.2.2 Functions

Functions play a crucial role in modeling the relationship between input and output populations in machine learning. Various types of functions, such as activation functions, loss functions, optimizer functions, evaluation functions, and regularization functions, impact the overall performance of machine learning models and influence the accuracy and efficiency of predictions.

Activation functions determine node activation, enabling neural networks to learn complex patterns. Sigmoid, tanh, ReLu, and SoftMax are commonly used, based on problem and architecture. Linear functions suit regression, while nonlinear functions are for classification. Sigmoid is for binary classification, ReLu for complex problems in hidden layers. Output layer's activation depends on problem type. Loss functions measure performance by minimizing predicted and actual output differences. Binary cross-entropy for binary classification, categorical cross-entropy for multiclass, MSE or MAE for regression. Choosing an appropriate loss function impacts accuracy and convergence. Regularization prevents overfitting and enhances generalization. L1 and L2 regularization add penalty terms. L1 penalizes absolute weights, L2 penalizes squared weights. Hyperparameters control regularization level, optimized via cross-validation. Optimizers adjust model parameters to minimize differences. Common optimizers include SGD, Adam, RMSprop, and Adagrad. Choice depends on dataset size, complexity, and training speed. Evaluation functions assess effectiveness, detect overfitting/underfitting, and commonly include accuracy, precision, recall, F1-score, and ROC curve. The choice of evaluation metrics depends on the model's objective, necessitating careful consideration and testing to optimize performance.

3.3 Limitation from Traditional Machine Learning in Semiconductors

From the experimentation carried out by the researcher in [38], extensively explored these traditional models but encountered significant constraints and limitations during experimentation. These findings served as a driving force for the researcher to look for alternative approaches that could potentially offer improved performance. During the course of the investigation, the researcher experimented with various machine learning models but found their performance to be inadequate for addressing the complexities of the semiconductor industry. However, the introduction of the autoencoder yielded more promising results. The autoencoder exhibited improved performance and demonstrated the potential for addressing the research problem more effectively. The successful implementation of the autoencoder in the semiconductor industry raised the question of whether neural network modeling could offer even greater benefits. As a result, the researchers recognized the limitations of machine learning models and emphasized the need to explore neural network techniques as a potential solution. Therefore, the current study aims to address the limitations of machine learning models by focusing on neural network modeling. By leveraging the insights gained from the [38] experiment and further exploring neural network architectures and methodologies, the study intends to overcome the limitations of traditional approaches and enhance predictive accuracy and efficiency in the semiconductor industry.

3.4 Summary

- The semiconductor industry utilizes wafer testing to identify defect-free dies that can be packaged and shipped successfully. The wafers undergo sequential testing to ensure their suitability under various conditions and to analyze their overall quality. Through sequential testing, the wafers are evaluated under different conditions to assess their overall quality and suitability. This essential manufacturing step involves electrical probing of each die on a wafer, recording results in a wafer map, and selecting passing circuits for further processing. By detecting and addressing manufacturing issues, wafer testing ensures the maintenance of high quality standards.
- Machine learning is AI's emphasis on data-driven algorithm development and prediction without explicit programming. It entails training models on vast datasets to identify patterns, extract insights, and enhance performance over time. With neural networks and classification/regression functions, machine learning has gained popularity across domains for its exceptional performance and problem-solving abilities.
- The limitations of conventional machine learning methods have highlighted the potential of autoencoder, prompting a call for further exploration of neural network modeling techniques. The objective is to advance the comprehension and implementation of neural network models by utilizing insights from prior research to discover novel approaches and make meaningful contributions to the existing knowledge, ultimately improving the effectiveness of problem-solving endeavors.

4 Data Preparation

To build accurate and efficient model, data must be properly formatted, cleaned and stored in a way that allows for easy access and manipulation. As data preparation involves a series of steps that use popular techniques such as data cleaning, normalization, feature scaling, and dimensionality reduction. These tasks include identifying and handling missing values and outliers, with the goal of improving the quality of the data and making it more suitable for using in algorithms. This section provides and overview of the considerations and techniques involved in data preprocessing and data augmentation for the data used in this research study.

4.1 Data Pre-processing

Advancement in the field, it may be essential to conduct research on data pre-processing, which involves developing novel techniques and algorithms capable of managing complex and diverse datasets. Categorizing data prior to analysis is a crucial factor that can impact the final outcome of a machine learning model. Utilizing pre-processed data can lead to improved training as it requires fewer resources, resulting in a more efficient model. Creating established metrics and benchmarks is crucial for evaluating the effectiveness of various data pre-processing techniques which enables the identification of strengths and weaknesses of different methods, thus guiding further steps. It is vital to create techniques that are both scalable and efficient while maintaining optimal performance. Utilizing open source tools and libraries in data pre-processing can expedite the process and provide access to reusable, high-quality code. This enhances the reproducibility of research and promotes the rapid development of novel techniques and algorithms, ultimately propelling the field of artificial intelligence forward.

4.1.1 Data Cleaning

Expertise in data pre-processing is drawn from various domains. The appropriate techniques depends on the specific characteristics of the dataset and the machine learning algorithm's goals for analysis. In the data preprocessing phase, several tasks are performed, including data collection, cleaning, and splitting. One critical aspects of data preprocessing is handling null or missing values as missing data can significantly impact the algorithms performance. There are different strategies in handling missing data, such as imputation and deletion. Imputation method uses other available data in estimating the missing values. Replacing missing values with mean or median values or imputing them with values from similar data points may appear to be a quick fix, but this can lead to biased or inaccurate results, skew the data distribution, and potentially result in incorrect conclusions during analysis. Another method is deletion, removal of missing values for maintaining accurate data analysis and consistency in the provided dataset. According to the research [7][8], it is recommended to use the deletion method instead of imputation to handle missing values as many machine learning algorithms are incapable of processing missing data, which could result in inaccurate outcomes. On the other hand, removing null values ensures that the analysis is based on accurate and complete data, resulting in a smaller dataset that is more accurate, less prone to bias, and improves dataset quality.

The research data in this study consists of two distinct phases, namely the input data and the target data. The data collection process involves obtaining data from the NXP source, which is generated through carefully designed experiments. This data is specifically selected to be representative of the problem domain and contains all the necessary features and target variables needed for addressing problem domain. By collecting this data, we ensure that our analysis and modeling are based on relevant and meaningful information that accurately reflects the underlying problem that is being addressed.

The input data includes various tests conducted sequentially, and the results of these tests determine whether the dies have passed or failed. The data collected during this phases is presented in the Table 4. Similarly, the target phase data follows a similar structure, but it may exhibit slight variations in voltage and temperature. Nevertheless, the overall process remains the same. However, it is important to note that both phases of the data may contain missing values or empty columns represented as NaN. These missing values can introduce noise into the model and affect its performance. To address this issue, it has been decided to handle the missing values by removing the corresponding missing values rather than replacing the missing values. This approach is preferred due to the drawbacks associated with replacing missing values, as discussed earlier. By removing the values with missing values, the potential noise in the data can be mitigated, ensuring a cleaner and more reliable dataset for further analysis and modeling.

Wafer	Temperature	die_x	die_y	Test1	Test2	Test3	….	TestN	Hardbin	die_id
	125^{o} C								0	
	125^{o} C								0	
	…..								…	
	125^{o} C								1	

Table 4 Structure of Data

4.1.2 Feature Selection

Appropriate feature selection choice depends on various factors, such as type of data, the problem domain and the availability of resources. On the other hand, choosing the wrong model technique can result in poor performance, increased training time, and decreased model interpretability. The paper[7], highlights the importance of using appropriate feature selection techniques for achieving better performance in machine learning models. The significance of feature selection lies in its capability to eliminate redundant or noisy features that do not contribute to the model's performance. In accordance with [7], most commonly used classifiers were compared on the identical dataset. The metric for validation as for the study was the accuracy which results indicated that Random Forest and AdaBoost produced features that could give a good results compared to other feature selection techniques. The measure of accuracy represented the correlation between the selected features, the measure from other techniques produced features that were 83 percent related which resulted in average performance of the model from this techniques. Furthermore, the study recommends the random forest classifier for feature selection, as it demonstrated superior performance in the experiments on their identical dataset. From the pool of available variables, only two testing variables are chosen for their substantial influence on the model's prediction. These selected variables, referred to as test variable 'i' and test variable 't', are depicted in the Table 5, showcasing their respective data structure.

Data	Raw Data	Pre-processed data ('i')	Pre-processed data ('t')
X variable data	(5832, 12515)	(5832, 142)	(5832, 107)
Y variable data	(6086, 9705)	(6086, 142)	(6086, 107)
Merged (X&Y)	-	(6086, 139)	(6086, 104)

Table 5 Number of units before and after handling data

According to reference [26], selecting features using Random Classifier can lead to the selection of features that are more strongly correlated with each other, ultimately enhancing the model's performance. These insights prove more valuable to both researchers and practitioners seeking to identify the most appropriate technique for feature selection task. As for the random forest, it excels in capturing intricate relationships between the features by leveraging the ensemble learning capabilities, demonstrates efficient processing capabilities and can handle high-dimensional feature spaces. As a result, it emerges as an optimal choice for a wide range of applications with diverse requirements.

4.1.3 Feature Importance

We can evaluate the significance of each of feature and aggregate their results to make informed decision on feature inclusion or exclusion in the model. Feature importance[24] in random forest aids in selecting the most informative features, improving model performance, and reducing computation time. Feature scores are obtained using metrics like Gini impurity or information gain, and a subset of features can be chosen based on a threshold or the highest-ranked features. The researcher can decide on the approach, and the chosen subset becomes the final feature set for training and prediction. In this approach, feature importance is determined and ranked, providing insights into their relevance and significance. The random forest algorithm calculates reliable feature importance scores. Sorting them in descending order helps identify the most influential features as in Table 6. Top features are commonly selected based on their importance ranking. In this research, the top 100 features are chosen. This approach reduces dataset dimensionality[12], improves model performance, and enhances interpretability. Two options are considered: selecting either 100 or 70 features based on dataset size. The goal is to balance feature selection and model accuracy, finding the best combination for optimal predictive power with minimal dimensionality.

Features	Importance Score
Feature 34	0.01119
Feature 42	0.01054
Feature 56	0.01053
…	…

Feature 15	0.00325
Feature 139	0.00135
Feature 21	0.0

Table 6 Feature selection Importance

4.1.4 Feature Scaling

Feature scaling is a process that adjusts the range of features in a dataset to a standardized or normalized scale. The choice of scaling technique depends on the specific algorithms being used. Some algorithms, like decision trees, are not affected by feature scaling and do not require it. However, other algorithms, such as neural networks, may benefit from feature scaling to achieve optimal performance. The decision to apply feature scaling depends on the requirements of the algorithm. If scaling is necessary, the choice of technique depends on the distribution of the data. If the data follows a normal distribution, standardization (also known as z-score normalization) is commonly employed. This technique scales the data so that it has a mean of 0 and a standard deviation of 1. On the other hand, if the data is not normally distributed and contains outliers, normalization (also known as min-max scaling) can be used. This scales the data to a range between 0 and 1[15].

Initially considered for feature scaling due to the presence of outliers, the MinMaxScaler exhibited poor performance in validation. It scales data by subtracting the minimum value and dividing by the range (the difference between the maximum and minimum values), making it sensitive to outliers and less effective than the StandardScaler for features with high variance. In the presence of outliers, the data range becomes distorted, biasing the scaling and reducing the effectiveness of normalization[18].

The StandardScaler is widely used in machine learning to perform scaling, making it popular among practitioners. It transforms the data to have a mean of 0 and a standard deviation of 1, standardizing the features. This is advantageous for algorithms that assume a normal distribution or when variables have different scales. Normalizing the data reduces the influence of outliers, prevents variables from dominating computations, and leads to more balanced and reliable calculations. It simplifies data visualization and interpretation by aligning variables on the same scale, enabling easier comparison and analysis. Additionally, the StandardScaler reduces the impact of outliers, resulting in a more robust and reliable model. It is widely accepted and can be easily implemented in various machine learning libraries and frameworks. When dealing with outliers, using the StandardScaler is recommended over the MinMaxScaler, based on the specific needs and dataset properties. It is important to perform feature scaling after splitting the data into training and testing sets to prevent information leakage. Proper feature scaling ensures consistent scales and avoids potential biases or issues arising from feature range differences.

4.2 Data Augmentation

Data augmentation plays a crucial role in enhancing the quality and quantity of the training data during model training. It involves applying various transformations to the original data, thereby creating new artificial samples. The primary goal of data augmentation is to increase the availability of training data, leading to improved model performance and generalization capabilities. This technique is particularly valuable when working with limited amounts of data, as it helps mitigate overfitting issues and enhances the model's robustness. Common augmentation techniques include rotation, random translation, mix-up, and SMOTE. In addition to expanding the training data, data augmentation aids in improving the model's ability to generalize to unseen data. By exposing the model to a diverse range of data, it learns to identify relevant patterns and features for the given task, rather than solely relying on memorizing the training samples.

Data augmentation is a valuable technique for enhancing the training dataset by increasing its size and diversity. In practical scenarios, the available dataset is often limited in size and may not adequately represent the full range of variations and complexities present in the target population. Data augmentation addresses this limitation by generating additional samples that resemble the original data but with certain modifications. This process enables the model to learn more robust and invariant features, reducing the risk of overfitting and enhancing performance on unseen test data. Additionally, data augmentation helps address the challenge of class imbalance by generating new samples for minority classes. This aids in improving the recall and precision of the model, leading to better overall performance.

When working with large data sets, it is important to thoroughly investigate and explore the data, identifying correlations between columns. This could entail creating new methods for dealing with imbalanced data and outliers. To facilitate easier training and eliminate bias towards any particular class, it is essential to ensure a balanced data set. In this particular context, the research paper [7] identified the SMOTE technique as the most suitable approach to be applied to the dataset. SMOTE generates artificial instances of the minority class by interpolating between existing samples. By randomly selecting minority class samples and identifying their nearest neighbors, SMOTE creates new synthetic samples along the line segments connecting them. This process helps balance the class distribution and provides more representative training data for the minority class. SMOTE is particularly useful for models that are sensitive to imbalanced class distributions and can improve their performance.

Class	Without Augmentation
0	5343
1	743

Table 7 Class difference in dataset

The Table 7 presented indicates a considerable imbalance between the classes in the data used for this study. Given the recognized effectiveness of SMOTE in handling imbalanced datasets, as demonstrated and validated in the reference paper [7], this study employs SMOTE as a technique for generating additional samples. The below data class has 0 as the positive class and 1 as negative class.

Following the division of the input and target populations into training and testing sets, data augmentation is applied exclusively to the training data, while leaving the validation and testing data unchanged. This approach ensures that the newly created samples are used exclusively for training, preserving the reliability of the validation and testing procedures.

TrainTest Split	Class	Without Augmentation	Applied augmentation	Before Augmentation	After Augmentation
y_train	0	3870	3870	4381	7740
	1	511	3870		
y_val	0	953	-	1096	1096 *(same)*
	1	143	-		
y_test	0	520	-	609	609 *(same)*
	1	89	-		

Table 8 Structure of Data difference applying SMOTE techniques

By integrating data augmentation technique during the training process, the neural network benefits from enhanced pattern recognition and increased sensitivity to subtle details within the data. The augmented samples as depicted in Table 8 serve as valuable additional information, enabling the model to generalize better and make more precise predictions. This improvement in performance is attributed towards the heightening the robustness and adaptability of the neural network model to a wide range of data scenarios. A thorough exploration and understanding of the model's design and configuration, which is customized to suit the specific requirements of the task, are extensively discussed in Chapter 6, providing comprehensive insights into its capabilities and intricacies.

4.3 Summary

- Before training the model, data preprocessing is conducted, including the removal of missing or null values. Feature selection is then carried out using the Random Classifier algorithm, where the top 100 features are selected based on their importance. The data is further scaled using StandardScaler and MinMaxScaler methods in accordance with the model’s requirements.
- To address the issue of imbalanced class distribution in the data, data augmentation techniques are applied, specifically using SMOTE (Synthetic Minority Over-sampling Technique) on the training dataset. This helps to generate synthetic samples and balance the distribution of classes in the dataset.

5 User Interface

User interface serves as preliminary point for users and software application or system serving as the gateway for user to access its functionalities through visual and interactive elements, allowing users to navigate, input data and obtain corresponding results. A well-designed user interface is important in ensuring a positive user experience, enhancing usability and facilitating efficient interaction with the system. Recognizing the multifaceted significance of the user interface, this chapter delves into the design aspects of a user interface specifically tailored to facilitate easy interaction with the machine learning model. By ensuring a seamless and engaging user experience, this designed user interface optimizes the efficiency of the model's processes and enhances overall user satisfaction.

5.1 Python Flask API

Python Flask is a popular web framework for developing and deploying applications, especially when combined with machine learning. It enables the creation of lightweight APIs that incorporate machine learning functionality and interact with software components. The Flask-based system loads a pre-trained model, processes incoming data, makes predictions, and returns results through the API, integrating machine learning into web applications and data analysis pipelines.

The paper [41] explores the utilization of a framework to create a local user interface for a machine learning model. In this study, Django is the chosen framework, known for simplicity and minimalism, aiding web app development. Python, Flask enhance functionality and ease of use, forming a powerful foundation for ML-enabled web apps. Flask, with minimal dependencies and flexible customization, stands out among web frameworks, making it ideal for researchers with specific requirements and offering seamless integration with Python libraries for machine learning tasks. The paper also highlights Flask's features, including route handling and response formatting, make it a compelling choice for building RESTful APIs, supported by a thriving community.

5.2 Database

Alignment of Python Flask API with a Database involves establishing a connection between the API and the database to facilitate seamless data retrieval and storage. API integration facilitates CRUD operations on the database, while the alignment process entails multiple considerations:

1. Choosing the appropriate database: Selecting a suitable database that meets the application's requirements. In this study, the Microsoft database management system (DBMS) is used.
2. Configuring the database: Adjust settings and create a data model, including migrating data to the database schema.
3. Establishing the connection: Set up the Flask application to connect with the DBMS using connection parameters and firewall settings.
4. Defining API endpoints: Create Flask endpoints for CRUD operations, enabling interaction with the DBMS.

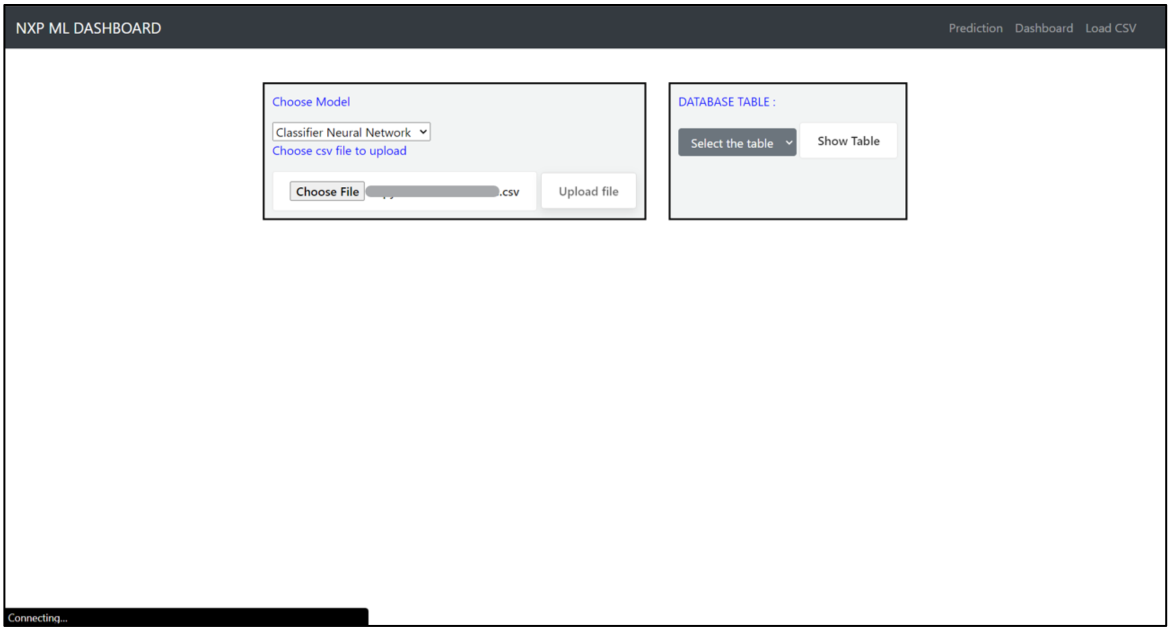
5. Utilizing the ORM library: Utilize an ORM library like SQLAlchemy to fetch and manipulate data, making it compatible with machine learning tasks.

By aligning Python Flask API with a DBMS, the app can utilize a database's power to store and fetch data effectively. This empowers the API to offer dynamic and data-driven features. The user interface displays preprocessed data from the database, improving performance by reducing real-time computations and response times, especially with large datasets. It conserves system resources and ensures data consistency. By offloading data preprocessing to the database, the interface simplifies development and maintenance. Leveraging a DBMS provides direct data access, real-time updates, efficient querying, data integrity, scalability, and integration with database features. However, loading a CSV file as input is more convenient for users unfamiliar with the DBMS platform, simplifying the process. It's important to consider the trade-off between convenience and the advantages of a DBMS. The user interface integrates smoothly with backend machine learning procedures, including data validation, preprocessing, real-time results, visualizations, customization, error handling, and continuous improvement. This enhances usability, efficiency, and user satisfaction.

5.3 User interface design

The user interface (UI) enhances usability through a well-designed, intuitive interface that simplifies tasks, reduces learning curve, and increases user satisfaction. It improves productivity, saves time, fosters loyalty, minimizes errors, and promotes familiarity with consistent design. The Python Flask API and HTML are utilized in the interface design to create an interactive and feature-rich user experience. Machine learning capabilities are harnessed through TensorFlow and scikit-learn packages. Routes are defined using the *@app.route* decorator, determining the URL path for each route. The view function, placed below the decorator, is executed when the corresponding route is accessed. The '*/form*' handles both GET and POST requests, rendering an HTML form for GET and retrieving form data for POST. The *if __name__ == '__main__'* block ensures the Flask app runs only when the script is executed directly. Flask offers additional functionalities like form handling, database integration, and authentication.

Various functions in the design use defined functions to process inputs, execute algorithms, and generate HTML outputs. These outputs offer visually appealing and informative interfaces with desired information or predictions. The interface provides flexible input options: upload a CSV file or fetch data from the database according to specific preferences, as shown in the Figure 5. Pyodbc connects the interface to databases, simplifying communication and ensuring secure data retrieval. The obtained data is seamlessly displayed as an HTML table, enhancing the interface's functionality. The interface captures and saves the data as a input parameter, and subsequently employs preprocessing techniques for analysis, as mentioned in Section 4. The resulting processed data is then used for machine learning tasks within the interface.



Choose Model: Classifier Neural Network

Choose csv file to upload: [redacted].csv

Figure 5 User Interface

After preparing the data through preprocessing and utilizing saved models, the relevant features are chosen and combined with pre-scaled models and pretrained machine learning algorithms. This combination allows for precise predictions on unseen data. The results, presented in a user-friendly interface with a table format, effectively convey the predicted outcomes, thereby improving decision-making based on machine learning.

The pre-trained machine learning model, along with the scaler and list of selected features, are saved as h5 and pkl models to ensure their preservation and reusability within the system. This allows for easy retrieval and integration of the models into the interface, providing seamless and efficient predictions By opting to save the model in widely used formats like h5 (Hierarchical Data) and pkl (pickle), it becomes effortless to load and employ the model whenever required. Storing machine learning applications in these standardized file formats ensures convenience and compatibility, allowing for seamless integration and utilization of the model as needed. The models allows for the preservation of the trained model's state, enabling the interface to perform predictions without the need for retraining. It provides a convenient way to store and share trained models, facilitating reproducibility and collaboration. It also helps in maintaining consistency of the results, as the same model can be loaded and used by different users or in different environments. This consistency ensures that the same input data will always yield the same output predictions or results, regardless of the platform or user. Moreover, storing models in standardized file formats simplifies the deployment process, as the model can be easily transferred and deployed on various systems or cloud platforms. This flexibility allows for scalability and efficient utilization of computing resources.

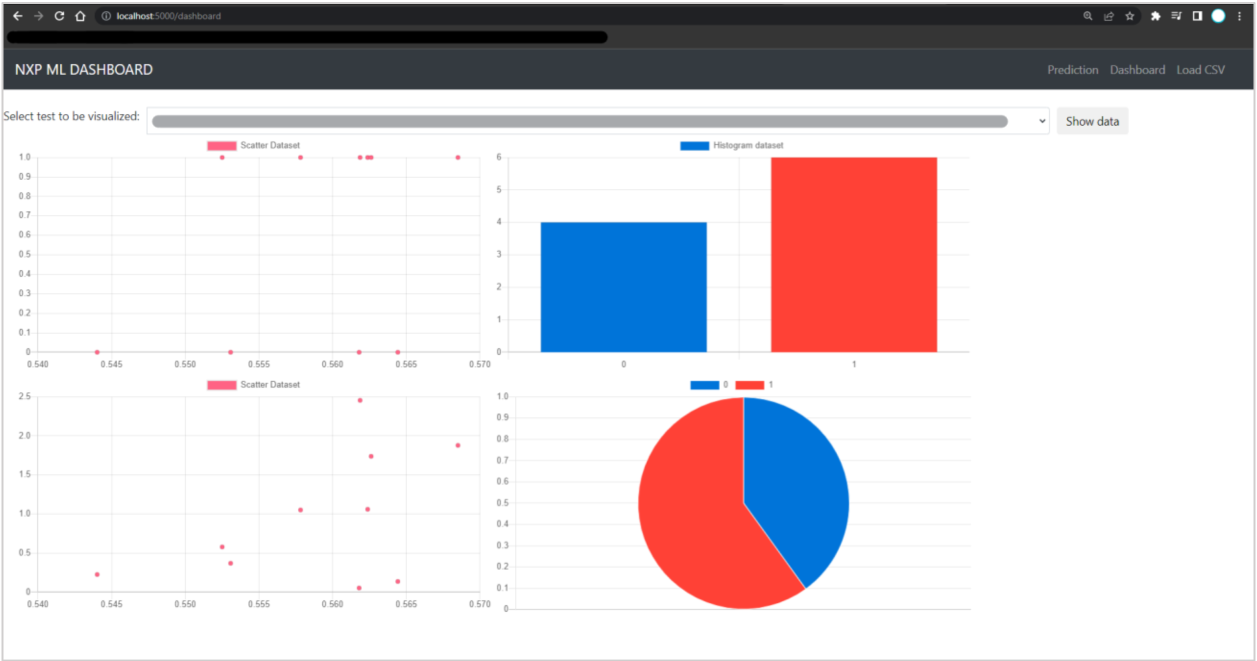


Figure 6 User Interface Visualization

In addition, the interface incorporates a graph visualization function as depicted in Figure 6 to present data in a graphical format. This feature allows users to gain a visual understanding of the data patterns, trends, and relationships. The graph representation enhances data analysis and provides a more intuitive way to interpret and communicate the insights derived from the machine learning models. The graph feature adds an interactive and visually appealing element to the interface, facilitating a more comprehensive and engaging data exploration experience.

The application is hosted on the local server, allowing users to access it through their web browser. The interface is aesthetically designed using CSS (Cascading Style Sheets) to enhance its visual appeal and provide a consistent and engaging user experience. The CSS formatting includes the use of colors, fonts, layout, and other styling elements to create an attractive and user-friendly interface. This attention to design details contributes to the overall usability and professionalism of the application.

5.4 Summary

Flask API simplifies real-time machine learning predictions, making it accessible to non-programmers. By leveraging Flask, developers create an intuitive interface for users to interact effortlessly with ML models. It streamlines integration, enabling immediate predictions and empowering non-technical users to gain insights without extensive programming knowledge.

6 Validation

In the validation phase, the aim is to explore the outcomes resulting from the utilization of a neural network model for predicting product characteristics in the semiconductor industry. This is accomplished by implementing the proposed architecture model in real-world context and examining the resulting effects. This process corresponds to the validation phase in the design science engineering cycle, where a single case-mechanism experiment is conducted to assess the performance of the validation model when applied to a specific subject of study. In this scenario, the validation model is based on the architecture created on the processed data discussed in the previous chapter. This architecture interacts with the intended context, which involves predicting product characteristics.

Furthermore, in order to assess the performance of the validation model, R.J.Wieringa (2014) emphasized the need to define measurement variables and the scale to be used. By specifying the measurement variables used in the validation process, this research aims to revisit its main objective. The objective is to predict product characteristics and streamline the testing process through the use of neural network modeling. The proposed architecture is examined to determine the extent to which it aligns with the defined goals and fulfills the identified requirements for implementing neural network-based prediction. In this section, a set of hypotheses is formulated to provide a basis for the prediction process using the neural network architecture examined in this research, specifically in relation to the reference architecture. These hypotheses serve as the underlying assumptions for investigating the performance and effectiveness of the neural network model in making predictions.

6.1 Hypotheses

In this study, five hypotheses have been formulated to address specific research questions and test the effectiveness of the proposed solution, as they provide structural framework for examining the research objectives and offer a systematic approach for assessing the impact and performance of the solution. The hypotheses focuses on validating the proposed architecture against the requirements defined as the objective of this research. It aims to measure the extent to which the proposed architecture satisfies these requirements. To investigate this aspect, a research method involving a single case mechanism experiment is employed. This method allows for exploring the response of the proposed artifact in its intended context by building and applying the artifact to a real-world scenario.

By formulating these hypotheses the research endeavors to generate evidence-based insights and derive significant conclusions that contribute to the existing knowledge and comprehension of the problem domain. Through meticulous analysis and interpretation of the data, valuable insights can be extracted, validating existing theories and offering fresh perspectives by illuminating the underlying mechanisms and relationships within the problem domain. The research findings have the potential to inform decision-making, stimulate further investigations, and ultimately propel the field forward.

1. **The neural network model will exhibit superior accuracy in predicting product characteristics when compared to traditional statistical models.**

From the reference architecture, the first key requirement highlights the importance of leveraging neural networks to capture complex patterns and relationships, leading to more accurate and reliable predictions compared to traditional approaches. Conducting comparative analysis for testing the performance between the neural network model and established statistical models with accuracy metrics such as mean squared error or accuracy rate can be used to assess and compare the prediction results. This will evidently show neural network outperforms in terms of accuracy providing strong evidence to support the requirement and highlighting the advantages of utilizing neural networks for prediction task.

Moreover, the hypothesis implies that the neural network architecture has the ability to effectively handle non-linear relationships and efficiently process large volumes of data, which contributes to the superiority of predictive capabilities[19]. As leveraging advanced computational techniques and sophisticated algorithms the neural network model will uncover intricate patterns and dependencies in the data that may not be easily discernible in traditional statistical methods. The outcome of the hypothesis will provide valuable insights of neural networks potential that it possess as powerful tool for prediction in various domains. Furthermore, these findings contribute to the existing body of knowledge by highlighting the comparative advantages of neural networks over traditional statistical models, particularly in terms of accuracy and performance[20]. The first hypothesis that was formulated suggests that the neural network model will exhibit superior accuracy in predicting product characteristics when compared to traditional statistical models.

2. **Increasing the size of the training dataset will lead to an improvement in the prediction performance of the neural network model.**

The second criterion is grounded on the understanding that larger training datasets[32] offer more information and patterns for the model to assimilate, potentially resulting in enhanced generalization and more precise predictions. To assess this requirement in a systematic manner, a step-by-step approach can be adhered. Initially, a baseline model is trained using a relatively small training dataset. The model's performance is then evaluated by measuring its prediction accuracy or other relevant metrics. Subsequently, additional data points are gradually added into the training dataset, systematically increasing its size. The model is retrained using the augmented dataset, and its performance is evaluated once more. We can assess the extent of prediction accuracy improvement and determine the suitable training dataset ratio by examining the model's performance across different ratios, as illustrated in the figure below. This analysis allows us to evaluate the impact of increasing dataset size and compare the depicted ratios in Figure 7.

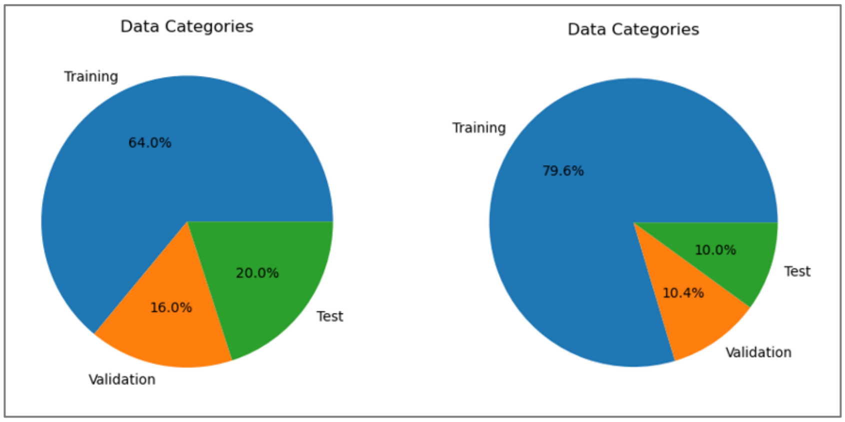


Figure 7 Train-test split ratio

Employing various statistical techniques and evaluation measures, the obtained results can be analyzed to assess their statistical significance. Through hypothesis testing, it is possible to assess whether the observed improvements in prediction performance are statistically significant or merely due to chance. If the results support the hypothesis, it will indicate that expanding the size of the training dataset indeed leads to an enhancement in the prediction performance of the neural network model. This finding carries practical implications, suggesting that incorporating more data into the training process can be beneficial for improving the model's accuracy and reliability in making predictions. However, it is important to carefully consider potential limitations and trade-offs associated with using larger training datasets, such as increased computational requirements and the risk of overfitting. Therefore, careful considerations should be made when determining the optimal dataset size for achieving the desired prediction performance.

The second hypothesis posits that Increasing the size of the training dataset will lead to an improvement in the prediction performance of the neural network model. This hypothesis stems from the belief that a larger training dataset enables the model to learn more diverse patterns and relationships, resulting in improved predictive performance.

3. **Including additional input features or variables into the neural network model will lead to improved performance metrics during the evaluation of predictions.**

The third criterion pertains to the integration by [7][12][29]into the neural network model, with the objective of enhancing its effectiveness. This entails introducing supplementary input features or variables to the model, thereby enriching the breadth and depth of information utilized for predictions. By enhancing the range of input data incorporated into the neural network, the model gains the potential to capture a wider array of relevant patterns and information. This augmented information empowers the model make more refined and accurate predictions, ultimately elevating its overall performance and predictive capabilities.

To assess the model's suitability for the given requirements, it undergoes a series of tests involving training and evaluation with various modifications to the input features. These modifications may include the inclusion of data sources, consideration of different aspects or attributes of the input data, or the introduction of additional contextual variables. Through rigorous experimentation and analysis, the model's performance can be compared across different configurations. The evaluation process entails assessing metrics such as prediction accuracy, precision, and the model's capacity to capture intricate relationships and patterns within the data.

If the outcomes of the analysis reveal a substantial enhancement in the accuracy and precision of predictions by incorporating input features, it will serve as supportive evidence for the third hypothesis. These findings indicate the potential benefits of expanding the scope of input variables, thereby enhancing the capabilities of the neural network model and enabling more precise predictions within the specified context. The third hypothesis involves the formulation of the idea that Including additional input features or variables into the neural network model will lead to improved performance metrics during the evaluation of predictions.

4. **Optimize the performance of the neural network architecture through the fine-tuning of its hyperparameters**

The fourth criterion pertains to enhancing the model's performance through optimization. This involves exploring and evaluating different combinations of hyperparameter values. The neural network model can be trained and assessed using various hyperparameter settings to measure their impact on the model's performance. The evaluation process entails analyzing metrics such as training loss, validation loss, and prediction accuracy. By comparing the outcomes obtained from various hyperparameter configurations, it becomes feasible to determine the settings that yield the most favorable performance. If the results indicate that specific values of the hyperparameters significantly improve the model's performance and lead to optimized predictions, it would provide support for the fourth requirement.

This would indicate that by fine-tuning the hyperparameters of the neural network [10][11], such as adjusting the learning rate, batch size, number of layers, and activation functions, it is possible to optimize the model's performance. The process of fine-tuning these hyperparameters enables the discovery of an optimal configuration that maximizes the neural network's predictive accuracy and generalization capabilities of the model's architecture, thus playing a crucial role in achieving optimal performance within the given context. The fourth hypothesis was formulated with the aim of optimize the performance of the neural network architecture through the fine-tuning of its hyperparameters.

5. **The neural network model will demonstrate generalizability by effectively predicting product characteristics on new or unseen datasets.**

The final criterion revolves around assessing the generalizability of the neural network model to new datasets, assessing its capacity to accurately predict product characteristics in unseen instances. To fulfill this requirement, the trained model will be tested using data that was not part of the training process. This unseen data will serve as a benchmark for evaluating the model's ability to generalize its learned patterns and relationships to new instances. The evaluation procedure entails inputting the unseen datasets into the model and comparing the predicted product characteristics against the actual values. By examining the accuracy and reliability of the model's predictions on these fresh datasets, we can gauge its proficiency in effectively applying its acquired knowledge to previously unseen instances.

If the model exhibits a strong level of precision and reliability in predicting product characteristics on unfamiliar datasets, it would validate the final criterion. This outcome would indicate that the neural network model has successfully captured and learned underlying patterns and relationships, empowering it to generate dependable predictions even in novel situations. By verifying the model's generalizability to perform well in diverse and unfamiliar datasets, we can strengthen our confidence in its capability to excel in real-life situations. This validation process will serve to authenticate the efficacy and practical value of the neural network model in accurately predicting product characteristics. The final hypothesis posits that the neural network model will demonstrate generalizability by effectively predicting product characteristics on new or unseen datasets.

6.2 Neural Network Models

This section, a thorough examination of the developed and analyzed neural network models, focusing on their model structure to ensure reliable prediction of product characteristics. The inclusion of both regression and classification models enables diverse insights and flexible analysis of different data types and problem domains. Regression handles continuous variables, predicting specific values, while classification handles categorical variables, predicting classes or categories. This combination broadens the scope of prediction tasks, accommodating various data and problems, improving overall predictive capabilities, and enabling comprehensive data analysis for informed decision-making.

The section outlines the models' methodology, and performance ability serving as a foundation for exploration of model architectures and their performance assessment, with particular emphasis on the regression and classification models known for their significance and versatility. This comprehensive analysis of the models' structure enhances our confidence in their ability to accurately predict product characteristics and reinforces the reliability of the results they produce.

6.2.1 Classification Model

6.2.1.1 Train-Test Split and Data scaling

The classification model is specifically designed to evaluate the performance of categorical output. The primary objective is to evaluate the model's ability to accurately classify the data and make predictions using binary categories. As described in Chapter 4, the dataset undergoes several preparatory tasks including data collection, cleaning, splitting, and augmentation to ensure balanced classes. The training set is used to train the classification model, while the validation set is utilized for hyperparameter tuning and model evaluation. Finally, the testing set is reserved for the final assessment of the model's performance after it has been trained using the designated model architecture. The train-test split is performed with various ratios, and it has been observed that using a ratios of 0.7 for training, 0.2 for validation, and 0.1 for testing yielded better results compared to other ratios [32]. This specific ratio allocation allowed for a sufficient amount of data for training, a reasonable validation set to fine-tune the model, and a smaller but representative test set for the evaluation of the model's performance.

To achieve optimal model performance, feature scaling is crucial. Neglecting feature scaling leads to biased and inaccurate predictions due to the model's sensitivity to feature scale. The StandardScaler is used for scaling features on the training data. While distance-based algorithms heavily rely on feature scaling, neural networks handle it well due to their capacity to learn complex relationships. However, scaling indirectly affects neural networks, causing issues like slow convergence, gradient problems, and biased weight updates. Activation functions such as sigmoid or tanh in neural network architectures are sensitive to input scale, resulting in some features having a larger impact than others. This can distort the learning process and harm overall performance. Applying feature scaling to input features addresses this issue. By transforming features to a similar scale, each feature contributes more evenly to the learning process. This allows for fair comparisons of weights and coefficients, enabling informed decision-making. Feature scaling prevents dominant features and achieves balanced representation, enhancing interpretation of feature parameters and resulting in reliable predictions. The correlation between the data is visualized in the heatmap displayed in the Figure 8, indicating their interdependence.

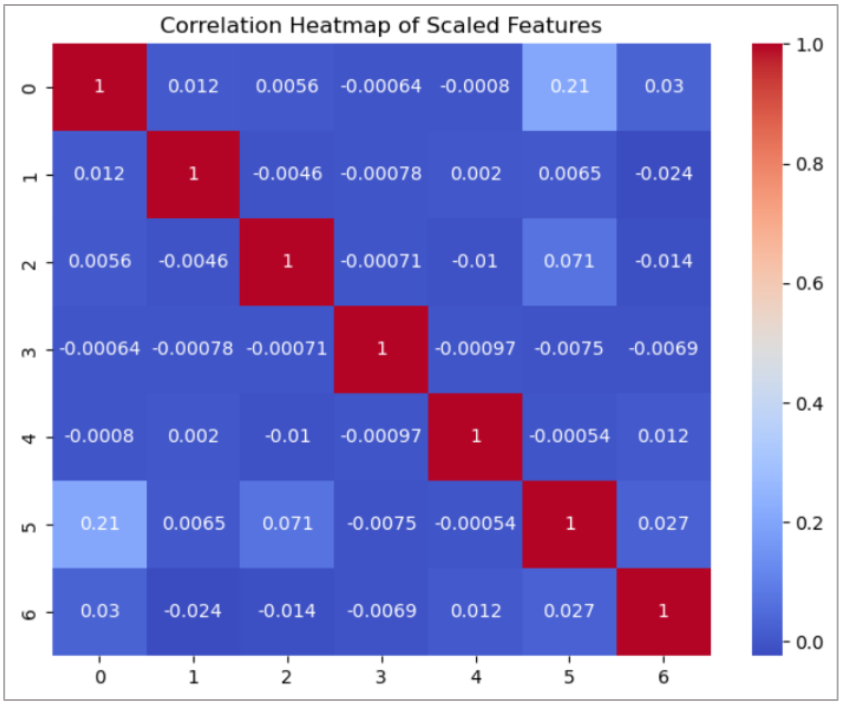
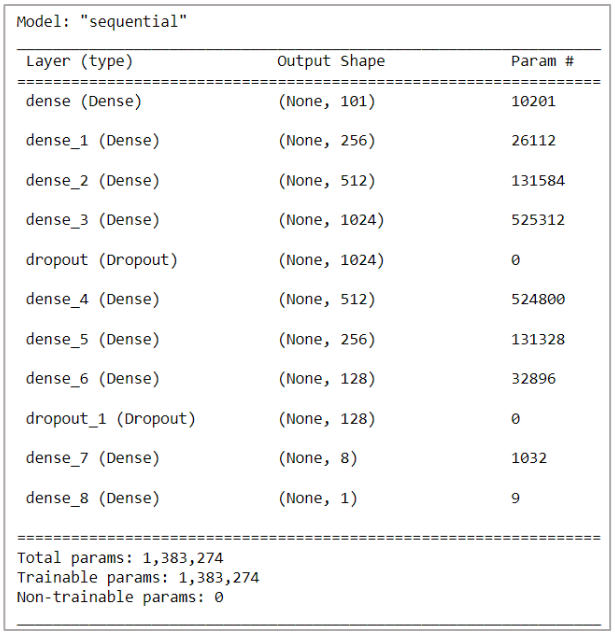


Figure 8 Correlation of Data

6.2.1.2 Design of Layers for Classification Model

The classification model design addressed the issue of imbalanced datasets. In the context of this research, the dataset exhibits a substantial disparity in the number of instances between the two classes. By balancing using SMOTE, the classification model became more robust and adept at handling imbalanced data and a deeper understanding of the underlying patterns in the data.

The classification neural network is carefully crafted to effectively process and learn from the input data. The initial layer, knowns as input layer, received the input data. The number of units is set to be equal to the input dimension, which represented the number of features or columns in the input data. Additionally, an additional weight was assigned to the input layer, resulting in a total of one plus the input dimension units in that layer. This configuration ensured that all input features are accounted for and processed by the neural network model. The activation function used in the input layer is typically the rectified linear unit (ReLu), which introduces non-linearity and enabled the network to capture complex mappings between the input and hidden layers[20]. The hidden layers, is positioned after the input layer, were responsible for extracting meaningful patterns and representations from the input data. By incorporating diverse activation functions and carefully determining the number of units and hidden layers, the neural network is crafted to effectively capture complex relationships and dependencies within the data. The classification neural network architecture incorporated multiple hidden layers with a specific configuration of units. The initial design of the hidden layer followed the reference architecture's formula for determining the number of layers and neurons[25][33]. However, this configuration resulted in poor performance and low accuracy. To address this, different layer configurations were experimented with until an architecture was found that could effectively train the model with complex patterns[14][20][34].



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 101)               10201

 dense_1 (Dense)             (None, 256)               26112

 dense_2 (Dense)             (None, 512)               131584

 dense_3 (Dense)             (None, 1024)              525312

 dropout (Dropout)           (None, 1024)              0

 dense_4 (Dense)             (None, 512)               524800

 dense_5 (Dense)             (None, 256)               131328

 dense_6 (Dense)             (None, 128)               32896

 dropout_1 (Dropout)         (None, 128)               0

 dense_7 (Dense)             (None, 8)                 1032

 dense_8 (Dense)             (None, 1)                 9

=================================================================
Total params: 1,383,274
Trainable params: 1,383,274
Non-trainable params: 0
_________________________________________________________________
```

Figure 9 Classification Model Input, Hidden and Output Layer Architecture

Typically, the number of units in each hidden layer follows a pattern where it is twice the number of units in the previous layer[34], gradually increasing until reaching a certain point, and then gradually decreasing as the network approaches the output layer as illustrated in Figure 9. This approach enables the neural network to extract increasingly abstract and higher-level features as the information propagated through the layers. The activation function chosen for all hidden layers is Rectified Linear Unit (ReLu). This activation function has demonstrated superior performance in terms of the model's decision-making capabilities compared to other options such as 'Elu' and 'tanh'. Although 'Elu' and 'tanh' showed reasonably good accuracy, their performance in other evaluation metrics fell short in comparison to ReLu. This underscores the effectiveness of ReLu in capturing complex non-linear relationships and patterns of the data.

The output layer is the final layer in the neural network architecture, responsible for producing the predictions or classifications based on the input data, specifically the desired class labels. The output layer of the network is configured with a single unit as Figure 10 represents and uses the sigmoid activation function. The sigmoid function squashes the output values between the range of 0 and 1, allowing for the interpretation of probabilities. By using the sigmoid function, the output layer is able to estimate the likelihood of a given sample belonging to a particular class, based on the output value it produces. For instance, a sigmoid output of 0.9 would indicate the higher probability of belonging to one class, while 0.1 for other class. In essence, the sigmoid facilitate the binary classification task by transforming continuous output of the neural network into probability representation allowing for the determination of class membership based on the probability threshold. The model is trained by processing the data through hidden layers, extracting useful information. This learned information is utilized in the output layer in determining the appropriate class label to the input data. This design is aiming in finding a balance between complexity and efficiency by creating a model that is able to effectively learn from the data and generalize well to new data, and accurately classify binary categorical outputs.

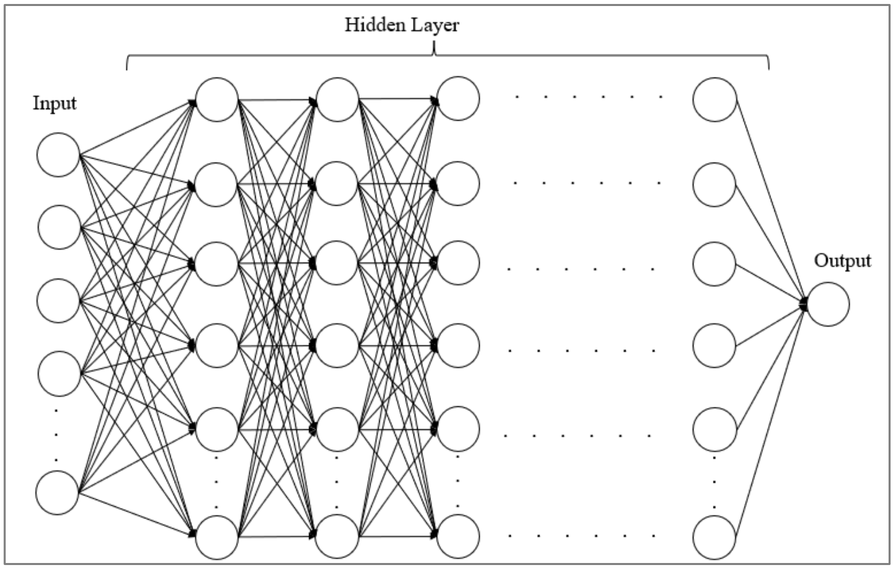


Figure 10 Graphical representation of Classification Neural Network

6.2.1.3 Overfitting

To avoid overfitting, the reference architecture incorporated dropout and regularization techniques. Dropout temporarily "drop out" certain neurons or randomly deactivates a portion of neurons during training iteration, which helps in preventing co-adaptation and encouraged the model to learn more robust features[35]. Whereas, regularization adds a penalty term to the loss function, discouraging large weights. The utilization of this function in the design aimed to enhance the model's ability to generalize and minimize the likelihood of memorizing the training data. By including dropout in the hidden layer[20], the model introduced a level of randomness that prevent it from fitting noise in the training data and encourage it learn by disregarding some representative features. Furthermore, the reference architecture also incorporated regularization methods, such as L1 or L2 regularization, which are explicitly mentioned within the experiment in [36] itself. Regularization and dropouts are powerful methods, but they need to be adjusted to align with data characteristics and requirements of the context. In implemented design, incorporating regularization in the hidden layer resulted in unsatisfactory performance. This was due to the excessive strength of the regularization, leading to an underfitting of the training data and difficulty in accurately capturing the underlying patterns. The poor performance observed can be attributed to the fact that regularization technique introduced excessive randomness and constraints in training. As a result, the model's ability to learn complex relationships was limited and it struggled in capturing important data features. On the other hand, introducing dropout after three hidden layers with the lowest possible dropout rate in the design prevented overfitting[20] and enabled the model to learn the desired patterns effectively[10]. The impact of incorporating dropout in the training model can be observed through a visual representation in the Figure 12. Additionally, Figure 11 demonstrates the model training without dropout.

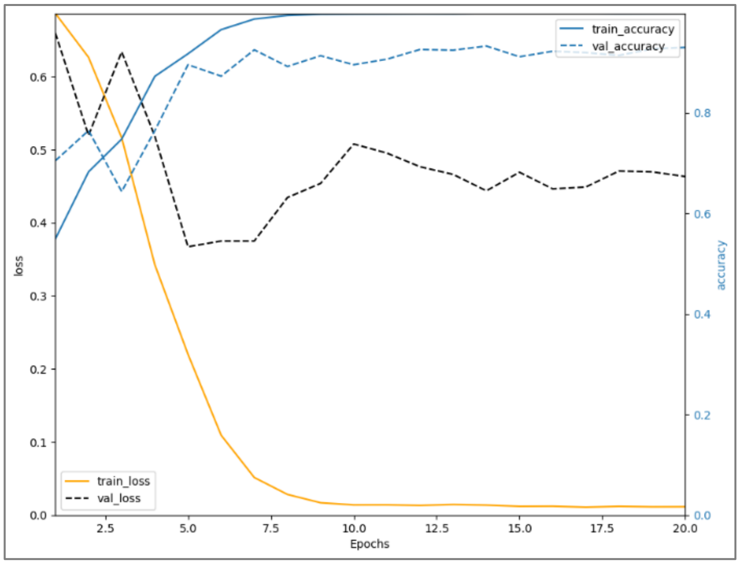


Figure 11 Model Training without dropout

In the experimentation, different dropout were tested in the hidden layers, parameters carefully chosen from the hyperparameter. As the 0.2 and 0.1 was finalized, it was observed that deviating from the specified values led to negative effects. Higher dropout rates caused model underfit with loss of learned patterns hindering the performance on new data. To ensure optimal results and assessment of reliability of model parameters, important to employ proper validation techniques that will monitor the performance that could determine the effectiveness of the chosen parameters and any potential issues. Figure 12 provides a visualization of the favorable outcomes achieved through the selection of appropriate parameters. Fine-tuning hyperparameters which acts as configuration settings for the model, through systematic exploration of different values allowed in identifying the combination that yield the most favorable results. This iterative approach allows researchers to continually improve and refine the model, achieving the best possible outcomes in the given context.

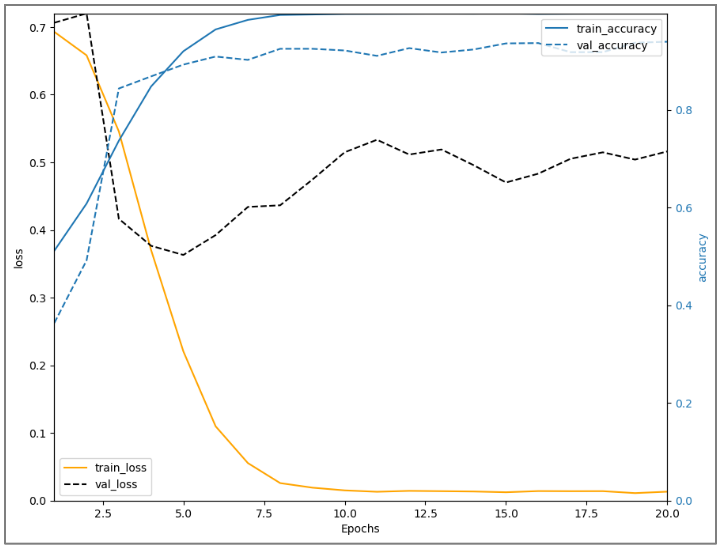


Figure 12 Model training with dropout

6.2.1.4 Hyperparameter Tuning

In the pursuit of optimal model performance, techniques like grid search are commonly utilized to identify the ideal set of hyperparameter values[11]. The effectiveness of a classification model is heavily reliant on the careful selection of appropriate hyperparameter values. These values have a significant impact on the model's performance, encompassing factors such as regularization, dropout and the structure of hidden layers (includes number of layers and units). Performing hyperparameter tuning[10], gives a good insight into the interplay between different parameters that can be examined. It highlights that if one hyperparameter[22] is optimized, but doesn't harmonize well with other parameters, it can compromise the overall model training. Right combination through experimentation and evaluation can simplify the relationship between them and maximize the performance metrics. It is important to note that the optimal hyperparameter values may differ depending on the specific dataset and problem being addressed. Additionally, exploring and experimenting with various regularization techniques and their combinations with other parameters in the model architecture, it was observed that the chosen regularization functions did not provide significant benefits based on the influencing factors considered. Therefore, it was determined that using regularization functions might not be the best choice for improving the model's performance in this context. Through different iterations and experiments with configurations and techniques that helped in refining, it was found that the optimal dropout rate values are 0.2 and 0.1 as they allowed the model to retain enough information and patterns. Through meticulous fine-tuning of the hyperparameters, such as modifying the dropout rates and incorporating pertinent techniques, a notable enhancement for model was designed. This process involved thorough analysis, experimentation, and fine-tuning to achieve the best results for the context.

6.2.1.5 Parameters

The chosen optimizer for the model design is Adam Optimizer, which is derived from the reference architecture. It has been selected due to its impact on the speed of convergence and its ability to iteratively adjust the internal parameters of the model in minimizing the loss functions. Adam is widely used in various applications, making it a popular and reliable choice for optimization. The choice of optimizer in this research is driven by evaluating performance based on the dataset and the architecture requirement. When Adam compared with RMSprop (Root Mean Square Propagation)[35] optimizer which is an optimizer that adjusts the learning rate based on the gradients of the parameter. RMSprop demonstrated favorable performance in this research where there are sparse gradients, leading to faster convergence. It achieved this by updating the learning rate individually for each parameter, resulting in improved outcomes. Empirical evidence serves as another influential factor in favor of using RMSprop instead of Adam during the experiment. By employing the RMSprop optimizer, the classification neural network was able to leverage the benefits of adaptive learning rate adjustments, efficiently update the model parameters throughout the training process. As a result, it contributed in the performance, convergence and accuracy of the classification model design. The experiment demonstrated that the RMSprop optimizer was a valid an superior choice over Adam optimizer in this context.

Considering the research objective, the model predict pass or fail in a binary classification task using binary cross entropy as the chosen loss function. It assigns a probability value between 0 and 1 to each instance, calculating dissimilarity based on information entropy. Minimizing this loss maximizes the likelihood of accurate class labels given the model's predictions. The dataset in this research exhibiting variations in class distribution, dies that pass are labeled as one class, while the failing dies have different labels based on the specific reasons for their failure. However, as the research objective is focused on identifying the failing dies and passing dies. Therefore, it has consolidate the failing dies with their different labels into a single class during the preprocessing task. This consolidation helped address the class imbalance and prevents bias letting it capture the relationships between the failing and the good dies. Converting the multi-class distribution to binary simplified the problem, improving computational efficiency and avoiding unnecessary complexity. The choice of binary cross-entropy loss aligned with the classification task, efficiently distinguishing between two exclusive classes and aligning with the objective.

Regarding the other parameters in the architecture, such as epochs and batch size for the model training, the number of epochs is represented as the iterations in which the entire training dataset is passed through the model. The batch size[9][36] is referred to number of samples that were processed by the model before the weights were updated during each training iteration. Similar to epochs[32], the batch size impacted the speed and efficiency of the training process. The larger batch sizes allowed for faster training by processing more samples in parallel, but they required more memory and computational resources. On other hand, smaller batch sizes enabled for more frequent weight updates and potentially lead to better generalization, but they slowed down the training process. Insufficient epochs lead to an underfit model that failed to capture the intricate patterns in the data. Conversely, an excessive number of epochs resulted in overfitting, where the model became too specialized to training data and performed poorly on unseen or test data. Determining the optimal epochs and batch size involved balancing underfitting and overfitting, considering dataset size, model complexity, and computational resources. Typically, starting with a moderate batch size, it was adjusted based on performance and resource constraints. Monitoring validation performance guided the selection of the optimal number of epochs, finding the trade-off between bias and variance. Each epoch processed the entire training dataset, and increasing epochs extended training time but excessive epochs didn't significantly improve performance. Through experimentation and monitoring of the model's performance, the optimal combination for epochs and batch size is determined, resulting in a well-trained and efficient classification model. The success of training the model is contingent upon the design and structure of the model architecture. More intricate architectures often demanded more computational resources, leading to longer training times. Additionally, larger models which entailed with a higher number of parameters, necessitates a substantial amount of data to learn and converge.

The evaluation of the classification neural network model design involved assessing the metrics in the model to determine the efficacy of each parameter in the architecture. This evaluation process ensured that the designed model is capable of effectively achieving the research objective[30]. By conducting experiments and validation, the neural network architecture is refined in finding the optimal number of hidden layers and number of units in the architecture. This optimization process helped in determine the efficiency of the model in learning and classifying input data, thereby enhancing its ability to handle classification tasks effectively.

6.2.2 Regression Model

Regression model design entails on the capability of predicting continuous numerical values based on the given input variables. The design process encompasses various steps, including selection of variables, defining the model structure, estimating parameters, assessing performance and interpreting the results to derive meaningful insights for predictions[8]. The model is designed similar to the classification model but the model design is contingent upon evaluating model performance. Data cleaning and scaling procedure for regression are comparable to those used in classification model. However, data augmentation for regression is not employed since the model learn to predict continuous numerical value. The scaling techniques are applied to both the input and target variables ensuring optimal performance in training and prediction. In determining the most suitable scaling techniques for regression, various methods were evaluated. Specifically, the performance of StandardScaler and MinMaxScaler is compared and based on rigorous evaluation, the use of MinMaxScaler is found to provide appropriate results in assessing the model performance as the distribution of the data is transformed among them and scaled letting the model train on this data indeed gives good results on test. The heat map in the Figure 13 illustrates the correlations between the scaled data, demonstrating that they are interconnected.

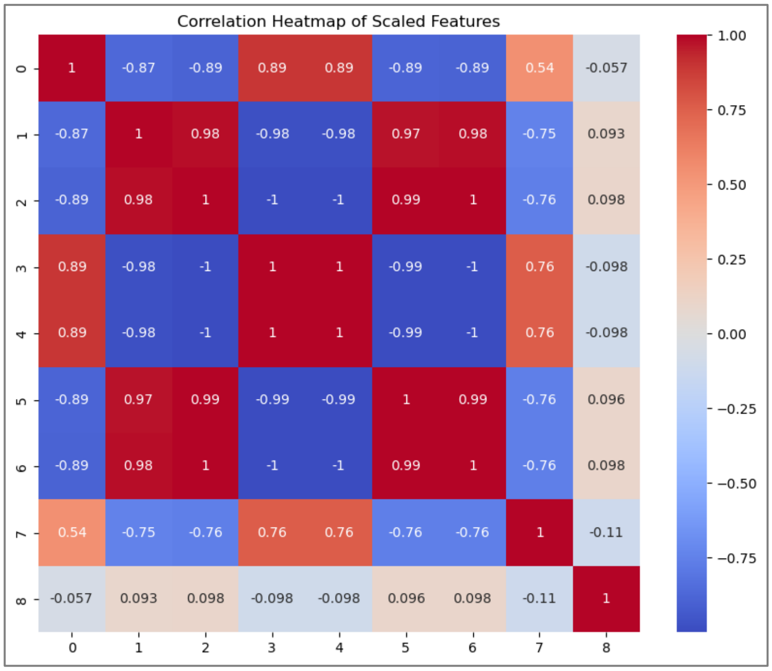


Figure 13 Correlation of Scaled Data

The model architecture exhibits slight variations compared to designed classification model. The variations include the use of a linear activation function in the output layer and having the same number for output units as input units as Figure 14. This design choice was to predict continuous numerical values for subsequent dies testing process and determine the corresponding predicted values. To evaluate the performance of the model, first step involved determining the number of hidden layers in the neural network architecture. The selection of an appropriate number of hidden layers directly impacted the model's capacity to learn complex patterns and represent the underlying data effectively. Additionally, dropout and L1 or L2 regularization are employed to mitigate overfitting in this model and improve generalization by reducing the models tendency to memorize training.

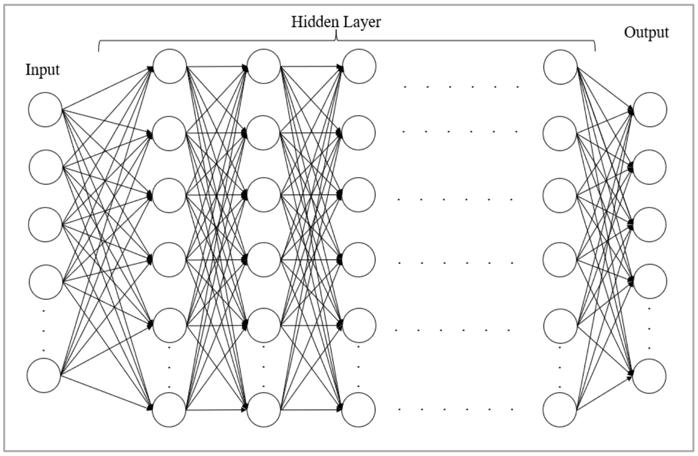
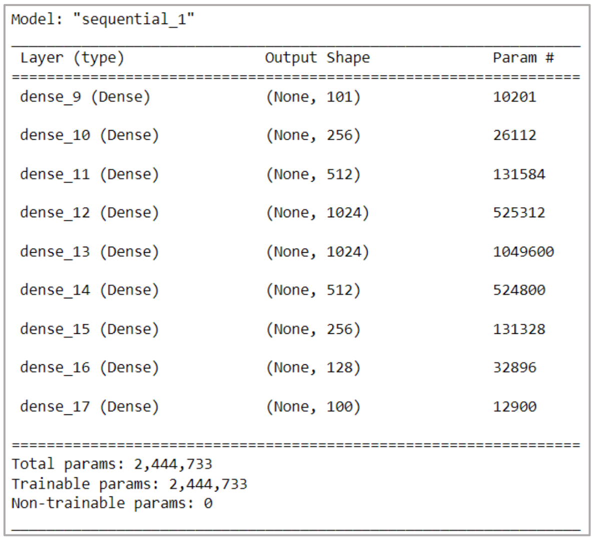


Figure 14 Graphical representation of Regression Neural Network

In the implemented design, dropout values of 0.5 and 0.2 are utilized, indicating approximately 50% and 20% of the neurons are randomly deactivated, respectively in the hidden layers[35]. Once the model is trained using the specified architecture and dropout values, its performance was rigorously evaluated to gain insights concerning the capabilities of the model. These insights played a role in making informed decisions about optimizing the model's architecture and ensuring its practical usability. However, it was observed that the model's performance fell short of expectations, indicating that the model is not effectively capturing the underlying patterns. This limitation was attributed to the chosen configuration of L1 or L2 regularization and also dropout, which result in the neglect of certain patterns. The model units of the layers are illustrated in Figure 15.



```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_9 (Dense)             (None, 101)               10201

 dense_10 (Dense)            (None, 256)               26112

 dense_11 (Dense)            (None, 512)               131584

 dense_12 (Dense)            (None, 1024)              525312

 dense_13 (Dense)            (None, 1024)              1049600

 dense_14 (Dense)            (None, 512)               524800

 dense_15 (Dense)            (None, 256)               131328

 dense_16 (Dense)            (None, 128)               32896

 dense_17 (Dense)            (None, 100)               12900

=================================================================
Total params: 2,444,733
Trainable params: 2,444,733
Non-trainable params: 0
_________________________________________________________________
```

Figure 15 Regression Model Input, Hidden and Output Layer Architecture

Early stopping techniques modify the duration of model training, monitoring the models performance on validation set. They prevent overfitting by halting the training process when the performance starts to degrade or reach a plateau. By terminating early, unnecessary iterations are avoided resulting in reduced training time. Implementation of early stopping in design for regression model, the training duration is effectively reduced. As model training would often requires extensive period, this technique helped in finding the point at which the models performance is optimal without further training. The length of model training with early stopping is determined by the value specified for the "patience" parameter. This parameter considered when to terminate training if the model's performance does not improve within a specified number of epochs. In regression models, the concept of patience was useful in determining the optimal stopping point during training by monitoring on validation set. It allowed the training to continue until there was no significant improvement or had minimal improvement in the validation metric. Setting the value of patience, had a significance for the number of epochs to wait before considering early stopping with effectively control training. The training process was halted when the validation metric had no improvement within the designated patience period which was effective in overfitting and yielded best possible results. Figure 16 displays the visualization of the training loss and validation loss for the model, revealing a smooth trajectory rather than a jagged path.

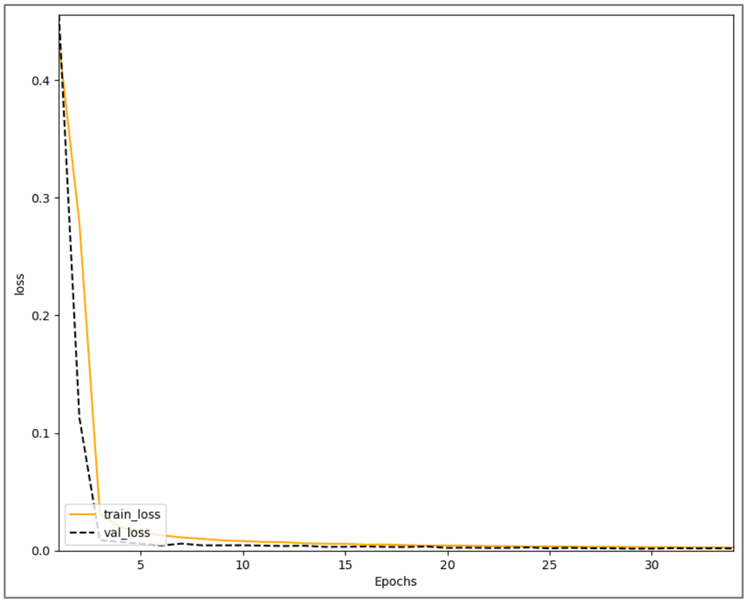


Figure 16 Regression Training, Validation Data Loss

On the other hand, in classification model, there was situations where using a patience-based approach for model training did not yield optimal results. This was especially evident as it was dealing with a complex architecture and an imbalanced dataset that required augmentation techniques to address the imbalance. In that cases, relying solely on patience as a stopping criterion in model training was not sufficient. The model design should be able to allow it to learn multiple layers and parameters, effectively capturing the distribution of different classes for the fixed number of epochs from the available data adequately. In contrast, regression models generally required less training time due to the simpler and less complex task of learning numerical relationships rather than task of discerning class boundaries.

The length of model training was affected by similar factors as those in the classification model. Specifically, the number of epochs and batch size played a role in models learning ability[36]. In this particular model, with careful consideration, a value of 700 epochs was determined that could yield the best results. This selection was based on the meticulous fine-tuning of hyperparameters that was able to address the context. Additionally, the batch size in conjunction with the epoch value is another factor that influence the effective of training. Through experimentation, it was determined that setting the batch size to 3/4th of the epoch size yield good results. As a result, batch size of 512 was identified as suitable value for the model.

For discrepancy between the predicted values and actual values, different optimization techniques is employed. In the implemented design, Adam and RMSprop were compared in terms of their abilities, evaluated based on monitor metrics and loss, particularly using mean squared as the chosen function. In the reference architecture, the choice of mean squared as the loss function proves suitable for minimizing the magnitude of error. This choice creates an incentive for the model to minimize the squared differences between predicted and actual values, which is particularly beneficial for the implemented design as a regression model. However, while RMSprop has shown promising results in classification, did not yield satisfactory outcomes in regression. Conversely, the Adam optimizer[26][34] demonstrated good performance. This indicates the choice of optimization for the algorithm have a significant impact on the success of the model emphasizing the importance of considering different options. Setting parameters as epochs, batch size and other parameters in this manner, the model was able to undergo an appropriate amount of training efficiently utilizing the resources and leveraging the inherent characteristics of the data. This approach ensured better alignment with the task, leading to satisfactory results.

As the architecture for regression neural network is determined, its performance is evaluated to assess its predictive power. This evaluation is based on metrics like mean squared error and RMSE[5][8], which measure how well the model fits the data[11]. Interpreting the results of the regression model for understanding the relationships between the independent variables and the predicted variables this involved the examination of the estimated coefficients for each variable which indicated their direction and strength of influence on the predicted outcome. Additionally, the reliability and generalizability of the regression model are ensured through validation[15]. This process involves applying the model to new or unseen data and assessing its performance on these data. As the model demonstrated good performance on the validation data, it instills confidence in ability to accurately predict outcomes.

6.3 Results

The main emphasis of this section is on the evaluation and validation of the implemented model's design performance. This comprehensive assessment encompasses multiple aspects, including effectiveness analysis, comparison with a reference architecture, and interpretation of the results obtained. The analysis involves interpreting the outcomes derived from the trained models, aiming to evaluate their accuracy in classifying unseen data and making predictions. To gain a deeper understanding of the model's strengths and weaknesses, the evaluation metrics are scrutinized. This entails analyzing its accuracy in correctly classifying different classes and assessing its performance on imbalanced datasets, when applicable. Additionally, we investigate the effects of hyperparameter tuning on the model's overall performance. Conducting this evaluation serves multiple purposes. Firstly, it enables drawing significant conclusions about the model's capabilities and constraints. Secondly, it facilitates making informed decisions regarding potential improvements and modifications to enhance its performance further. Lastly, by contributing valuable knowledge and insights to the field of neural network model design, this evaluation aids in advancing the current state-of-the-art and pushing the boundaries of what can be achieved in the realm of neural network modeling.

6.3.1 Model Performance

6.3.1.1 Classification Model

The performance and effectiveness of the classification model design is assessed using different metrics. Two metrics for evaluation are the AUC-ROC (Area Under the ROC Curve)[12] and the confusion matrix[10][32]. The AUC-ROC is a metric employed for assessing the overall effectiveness of the binary classification model. It will quantify the probability of the model assigning a higher score to a randomly selected positive instance compared to a randomly chosen negative instance. A higher AUC-ROC value[26] signifies stronger discriminatory ability in distinguishing between positive and negative instances, indicating superior performance of the model. Another method of evaluation involves utilizing the confusion matrix[10][32], which will provide a detail breakdown of the model's predictions, allowing for a deeper examination of its strengths and weaknesses. It will categorize the predicted values into four categories: true positive for correctly classifying of positive instances (TP), true negative for correctly classifying of positive instances (TN), false positive for classifying negative as positive instances (FP), and false negative for classifying positive as negative instances (FN)[30]. The matrix that will connect this four values with each other is given in Table 9. From the confusion matrix, various metrics such as accuracy, precision, recall, and F1 score can be estimated. These metrics provide insights about how well the model performs in correctly classifying positive and negative instances. By assessing the model’s performance using both AUPRC and the Confusion Matrix, a thorough comprehension of the model’s effectiveness in accurately classifying binary categorical outputs can be obtained. This evaluation provides valuable insights for refining and enhancing the classification model, ultimately leading to improved performance and results.

Actual Data		
Predicted	Non- Defective	Defective
Non-Defective	TP	FP
Defective	FN	TN

Table 9 Confusion matrix Layout

The implemented design for classification model was able to achieve an accuracy of 93% on the test data, accompanied with an AUC-ROC of 79%. These metrics demonstrate that the model is capable of making accurate predictions on the test data. To ensure low false positives, the specificity was measured using the confusion matrix. The specificity, known as the true negative rate, is a measure that quantifies the model's ability to correctly identify negative instances. The calculations are performed using the following formulas:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

The model exhibited a specificity of approximately 80%, indicating that it accurately classified negative instances around 63% of the time. This metric is essential in this scenarios because minimizing false negatives is vital. It reflects the model's capability to avoid misclassifying positive cases as negative. By carefully considering the specificity, the model's performance was evaluated, enabling informed decisions regarding the balance between sensitivity (true positive rate) and specificity within the given context. The Table 10 below provides a comprehensive overview of the performance of different evaluation of the classification model:

Different Parameters	Epochs-Batch Size	Accuracy	AUC-ROC	Precision	Specificity	Sensitivity
Test Attribute 'i'						
Dropout, Adam	500-1500	91.20	82.68	93.29	61.94	95.93
No dropout, Adam	500-1500	89.96	81.97	92.52	56.91	95.90
Dropout, RMSprop	500-500	93.30	84.32	96.19	72.44	96.19
Dropout, RMSprop	500-1500	93.26	83.95	97.11	80.76	95.10
No dropout, RMSprop	500-1500	92.66	83.52	95.62	69.30	96.03
L1/L2 regularization	500-1500	61.09	57.19	62.34	16.03	90.38
Test Attribute 't'						
Dropout, Adam	500-1500	91.62	83.45	95.00	71.11	95.18
No dropout, Adam	500-1500	90.64	81.94	94.23	67.39	94.77
Dropout, RMSprop	500-1500	94.08	84.43	98.07	86.30	95.14
Dropout, RMSprop	500-500	91.78	82.61	95.57	72.94	94.84
No dropout, RMSprop	500-1500	90.80	82.50	94.23	67.74	94.96
L1/L2 regularization	500-1500	87.11	49.57	99.15	00.00	87.76

Table 10 Classification Model Validation results measured in Percentage

By gathering the results from various evaluations conducted with multiple parameters, it becomes evident that the classification model performs exceptionally well when certain techniques are applied. Analyzing the Table 10 reveals that the classification model incorporating dropout regularization and utilizing the RMSprop optimization algorithm is the best. This combination yields the highest specificity value while maintaining a good AUC-ROC score and accuracy. These results suggest that employing dropout regularization and utilizing the RMSprop optimization algorithm contributes to improved performance and accuracy in the classification model, particularly in terms of specificity and predictive capability.

Furthermore, the statistical interpretation of AUC is straightforward: “if the classifier achieves an AUC well above 0.5, it is considered effective in identifying false positive instances. It would imply that the given valuable guidance on which modules should receive particular attention in testing, as stated in reference [30]. By examining the results and observing an AUC value well above 0.5, it indicates that the classifier is indeed providing valuable guidance on identifying instances that necessitate special attention during the testing process. This assessment allows for a clear understanding of the model's performance by considering two different test variables. It assesses the metrics of the model in making predictions while simultaneously minimizing false positive rates.

6.3.1.2 Regression Model

To obtain a comprehensive grasp of a regression model's performance, it is essential to take into account various evaluation metrics and techniques[8]. By thoroughly analyzing these metrics and referring to references[5][14], two specific metrics are employed to evaluate the model's predictive abilities, identify potential areas for enhancement[35], and make well-informed decisions regarding its suitability for the given context and dataset. The evaluation metrics for regression model differ from those used for classification models due to the distinct nature of the problem and the desired outcomes vary with each model type[34]. These evaluation metrics provide insights into regression model accuracy, and generalization capabilities. One such metric is the Root Mean Squared Error (RMSE) [16][29], which is square root of MSE[25][34]. MSE is obtained by averaging the squared difference between the predicted values and the true values[13]. The lower the RMSE, the better the models performance[15][23] in terms of minimizing the overall squared errors between predictions and actual values. Another metrics used is the Mean Absolute Error (MAE)[27], which calculates the average absolute difference between predicted and true values. MAE provides a measure of the average magnitude of errors, without considering the direction of deviations. The calculations are executed utilizing the subsequent formulas:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y_j})^2}$$

The y_j is the true value, $\hat{y}_j$ is the predicted value and n as number of observations/rows. RMSE will be 0 if the predicted value equals the true value, and the mathematical expression for Mean Absolute Error (MAE) is as follows[35]:

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y_j}|$$

After designing the regression model, the evaluation metrics indicated a RMSE of 0.034 and a MAE of 0.015. The evaluation results, including these metrics and other relevant parameters, are summarized in a Table 11. By analyzing the Table 11, it becomes evident that the model achieved its best performance when utilizing the dropout regularization technique and the Adam optimizer. The RMSE value of 0. 034 indicates that, on average, the predictions made by the model deviate by approximately 0. 034 units from the true values. Similarly, the MAE value of 0.015 suggests that, on average, the absolute difference between the predicted and true values is minimal, implying a high level of accuracy. By leveraging these parameters, the model can improve the prediction accuracy and minimize errors. The evaluation results highlight the success of the regression model in accurately predicting the target values, as reflected by the low RMSE and MAE values. The table provides a clear overview of the model's performance and allows in making informed decisions regarding the selection of parameters for future regression tasks.

Different Parameters	Epochs-Batch Size	MSE	MAE	RMSE
Test Attribute ‘i’				
Dropout, Adam	700-512	0.00374	0.03057	0.06119
No dropout, Adam	700-512	**0.00357**	**0.02931**	**0.05975**
No dropout, Adam	100-128	0.00367	0.03071	0.06058
Dropout, RMSprop	700-512	0.00669	0.03057	0.08180
No Dropout, RMSprop	700-512	0.00750	0.05736	0.08662
L1/L2 Regularization	700-512	0.14650	0.27734	0.38276
Test Attribute ‘t’				
Dropout, Adam	700-512	0.00461	0.04750	0.06790
No dropout, Adam	700-512	0.00163	0.01755	0.04047
No dropout, Adam	500-100	0.00165	0.01730	0.04070
Dropout, RMSprop	700-512	0.00438	0.04395	0.06625
No Dropout, RMSprop	700-512	0.00394	0.04752	0.06790
L1/L2 Regularization	700-512	0.31361	0.43298	0.56001
StandardScaler, Adam	700-512	1.31340	0.36157	1.14607

Table 11 Regression Model Validation results

6.3.1.3 Ensemble Model

The predictions performance is enhanced by employing an ensemble learning approach, which combines the prediction results from multiple base models[26]. This approach integrates the designed classification and regression base models, resulting in and overall model performance improvement. The approach leverages the diversity and characteristics of individual models, the ensemble generates more robust and reliable prediction[5]. The referenced paper[28], emphasizes that creating an ensemble of models in machine learning generates more stable and robust results compared to relying solely on the predictions of a single neural network model predictions and reducing the errors training. This approach leverages the collective wisdom of multiple models[20], enhancing the predictions. In this study, ensemble methods is constructed by summation of the predictions of both base models to create a consolidated prediction. Through the amalgamation of predictions, the ensemble model can benefit from the collective knowledge and strengths of the individual base models, leading to improved performance and more accurate predictions[27]. This results in improved predictive power, increased resilience against noise and outliers, and enhanced performance on unseen data[15][30].

The Table 12 summarizes the results obtained from an ensemble techniques on two base models: classification and regression neural network. It compiles the results achieved by combining the models predictions and using a defined threshold to determine the final classification output of the ensemble[20]. The combined predictive performance of the ensemble is presented in a concise format below, illustrating the overall results achieved.

Base model	Accuracy	AUC_ROC	Specificity	Sensitivity
Test Attribute ‘i’				
Classification(Dropout, RMSprop),Regression	87.35	83.95	77.27	87.73
Test Attribute ‘t’				
Classification(Dropout, RMSprop),Regression	89.32	83.86	90.0	89.29

Table 12 Ensemble Model Validation results in Percentage

6.4 Summary

- In hypotheses, the study outlines five assumptions that serves as framework for evaluating the capabilities, limitations, and overall suitability of the model in addressing the given problem and achieving the desired prediction outcomes. These hypotheses offer a structured approach to assess the model design, to enhance the performance and its alignment with the research objectives.
- Neural network model, comprising classification and regression modes, is elucidated to tackle the given problem by leveraging available data. This process involves exploring and assessing different parameter combinations in determining the configuration that maximizes performance based on predefined metrics. The identification of the optimal combination serves as a guide in constructing an effective model.
- Ensemble model improves upon the predictions made by individual base models, surpassing the performance of a single model's predictions.
- The neural network model's evaluation is presented in concise tables in the results section, summarizing various metrics like accuracy and RMSE. This organized format facilitates comparison, analysis, and interpretation of the model's performance, aiding in drawing reliable conclusions about its effectiveness.

7 Conclusion

In this section, a comprehensive synthesis of the research conducted on neural network models for prediction is provided, offering a thorough understanding of current state of the art. The primary focus is on the main objective of the research, which is predicting product characteristics using neural network models. The reference architecture for neural network modeling consists of key findings, insights and implication that serves as a guide for designing neural network model. These insights shed light on the underlying mechanisms and intricacies of the neural network, enabling a deeper understanding of their behavior and effectiveness in capturing and predicting product characteristics.

The structure of the research adheres to the design science research methodology, as described in the Section 1.5.2, which involves a series of steps. It begins with problem investigation and formulation of research questions. Followed by the treatment design, which includes specifying the requirements to achieve the research objectives, designing, and proposing an architecture for the neural network. The design of the neural network architecture follows the integrated reference architecture design, as described in a previous Section 2.5.3. The treatment validation is conducted using a single case mechanism, where designed model is tested on new data to assess their satisfaction with the proposed model's ability in meeting the requirements. This validation process ensures that the proposed model aligns with the intended goals and addresses the research objectives, providing confidence in its suitability and effectiveness.

7.1 Research Questions

- Which techniques can be employed to enhance the performance of neural networks for predicting anomalies in product characteristics and achieve zero defects screening strategies in the semiconductor industry?

SQ1. What are the most effective techniques for preprocessing product characteristic data for use in neural network training?

This study performs a systematic literature review to investigate the current state-of-the-art in neural networks. This review focuses on analyzing relevant scientific journal articles from the past decades to identify key components for enhancing product characteristics prediction using neural networks. The findings from the literature review, it is discovered that the adoption of neural network preprocessing techniques is primarily driven by the goal of enhancing data for model training and enabling reliable predictions. By leveraging these preprocessing techniques, researchers can enhance the overall quality and suitability of the data for neural network models. These techniques encompass various steps, such as handling missing values, scaling variables, addressing class imbalance, and other preprocessing procedures that contribute to optimizing the data for effective model training.

Researchers employed various strategies to deal with missing data, such as imputation or removing instances with missing values. Removing missing values stood out as one of the best methods for handling incomplete or unreliable data, ensuring that the model is trained on high-quality and meaningful information. Furthermore, scaling the variables using techniques like StandardScaler or MinMaxScaler, based on the problem's requirements, offers advantages in terms of feature standardizing and preventing dominance by features with larger magnitudes. This scaling process aids in achieving better convergence and enhance overall model stability.

In the case of class imbalance, a common challenge in classification problems, is effectively tackled using the SMOTE (Synthetic Minority Over-sampling Technique) technique. By generating synthetic samples for the minority class, SMOTE helps balance the class distribution and ensures that the model receives sufficient training data for the underrepresented class. This approach is well-regarded as an effective method for mitigating class imbalance issues and enhancing the model's performance in classification tasks.

During the feature selection process, various techniques like AdaBoost, PCA, and CFS were recognized from the literature, but the random forest classifier stood out as a particularly effective method. By using the random forest algorithm, researchers were able to determine feature importance, which played a crucial role in selecting the most relevant features. The algorithm's ability to handle complex relationships and evaluate feature importance provided valuable insights into which features had the greatest impact on the target variable. This facilitated the identification and selection of the best features for the task at hand, streamlining the dataset. The findings of the review suggest that the random forest classifier is a valuable tool for feature selection, enabling informed decisions and improving the accuracy and interpretability of predictions.

The findings from this literature review provide insights into the most effective preprocessing techniques for neural networks in the context of predicting product characteristics. By employing these preprocessing techniques, researchers can ensure that their neural network models are trained on high-quality, meaningful data, leading to more accurate predictions and improved performance.

SQ2. **What are the limitations and challenges associated with the use of neural networks for predicting in product characteristics?**

High flexibility leading to low interpretability, as they tend to overfit the training data and rely on complex interactions between features, making it challenging for domain experts to use and understand the model. Striking a balance between flexibility and interpretability is important for accurate and understandable models. Neural networks require abundant high-quality training data for precise predictions. However, they struggle with generalizing to new variations that were not encountered in training. This limitation arises because the network not being encountered to specific variations during training, and its ability to make accurate predictions in such cases became uncertain. To address this challenge, researchers carefully curated diverse datasets, employing data augmentation techniques, and applying appropriate regularization methods to improve the network's ability. By exposing the network to a broader range of data, including possible variations and outliers, it becomes more resilient and better equipped to generalize to unseen instances. These measures enhanced their ability to capture underlying relationships and make accurate predictions beyond the scope of the training data.

Developing and implementing neural networks demand expertise in data science and machine learning, and the model's configuration may be influenced by hardware limitations. The computational intensity of training and running neural networks increases with larger datasets and the number of neurons increases, impacting both accuracy and computation time. Selecting the optimal neural network architecture and setup can be challenging, as it would require careful consideration of the problem requirements.

SQ3. How can neural network architecture and design be optimized to improve the accuracy of predicting product characteristics for zero defects screening strategies?

To predict product characteristics accurately, it is essential to optimize the neural network architecture. This optimization entails making informed choices about the appropriate number of layers, neurons, and activation functions that are most suitable for the specific problem. Through experimentation, researchers can systematically explore different parameter options, such as learning rate, batch size, regularization techniques, and optimization algorithms, in determining the impact on the model's performance. By carefully evaluating and comparing these parameters, researchers can leverage the strengths of different architectures and improve the accuracy of predictions. This experimentation process is discussed briefly in Chapter 6 (Section 6.2), highlighting the importance of various parameters influence on neural network’s ability for achieving optimal neural network performance.

Furthermore, to enhance the accuracy of predicting product characteristics using neural networks, ensemble is employed on base models. Ensemble model involve combining predictions from multiple base neural networks to make a final prediction. This helps mitigate the impact of biases and limitations inherent in individual models, leading to improved overall accuracy. By leveraging the strengths of different models, ensemble model enable more reliable and robust predictions. This approach can be particularly beneficial when dealing with diverse or noisy datasets, as it improves generalization and prediction performance. Nonetheless, by effectively leveraging ensemble methods and combining predictions from base neural network model, researchers and practitioners can elevate the accuracy and reliability of predicting product characteristics, leading to more informed decision-making and improved outcomes in various domains.

Fine-tuning the model based on the specific requirements of the product characteristics problem can improve its performance and mitigate limitations such as overfitting or underfitting. This involves continuously monitoring the model's performance, retraining or updating it as needed, and optimizing its architecture and parameters, the neural network can maintain its effectiveness in predicting product characteristics. This iterative process ensures that the model remains accurate, adaptable, and aligned with the evolving nature of the dataset and product characteristics.

By carefully optimizing the neural network architecture and design through these approaches, it is possible to enhance the accuracy of predicting product characteristics in zero defects screening strategies. It is important to experiment, iterate, and fine-tune the models based on the specific requirements of the problem and the available data.

SQ4. How can the performance of the optimized neural network be validated and continuously used in real-time operation?

In order to validate the neural network, various metrics can be employed to measure its prediction capabilities. The specific choice of metrics depends on the type of neural network model being designed. For classification models, accuracy is a commonly used metric to measure the overall correctness of the model's predictions. It represents the ratio of correctly classified instances to the total number of instances in the dataset. Additionally, the confusion matrix is often utilized to gain deeper insights into the model's performance. It provides information about the number of true positives, true negatives, false positives, and false negatives, allowing for a more detailed analysis of the model's ability to correctly classify instances and the distribution of its class predictions. Using a confusion matrix, the performance of a neural network model can be assessed based on false positives and false negatives, and subsequently minimize them by calculating specificity and sensitivity. Specificity measures the model's ability to correctly identify actual negative cases, while sensitivity measures its ability to correctly identify actual positive cases. By closely monitoring and optimizing both specificity and sensitivity, a balance can be achieved in minimizing false positives and false negatives. This balance is crucial in various applications, where both types of errors can have significant consequences. Analyzing the confusion matrix and performance metrics allows researchers to comprehensively evaluate the model. Areas of improvement, such as sensitivity and specificity, can be identified to minimize false prediction, enhancing the neural network's overall predictive accuracy.

For regression models, metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are commonly used to validate the model's performance. These metrics assess the magnitude of differences between the actual and predicted values, allowing for a comprehensive evaluation of the model's overall performance, where smaller values indicate better accuracy. It is important to consider these metrics collectively to assess the performance of regression models for predicting product characteristics. By analyzing these metrics, researchers can determine the model's accuracy, precision, and ability to generalize to new data, allowing for informed decision-making and further improvement of the predictive capabilities of the model.

To facilitate the real-time operation of the model, it can be beneficial to develop a user interface that incorporates the trained neural network models. This interface would allow users to input data files in specified format and utilize the models for making prediction. By simplifying the prediction process and providing a user-friendly interface, users can effortlessly harness the capabilities of the neural network models without requiring in-depth knowledge of the intricate technical aspects involved.

7.2 Limitation and Future Research

Based on the findings of this study on the use of neural networks' for predicting product characteristics, a number of limitation is identified for future research is the absence of a precise mathematical formula to define the optimal architectural components for training the model on a given dataset size. Neural networks possess high flexibility and adaptability, enabling them to capture complex data relationships. However, this flexibility also hinders the ability to determine the exact architectural specifications, such as the number of layers, neurons, and their connections, solely through mathematical formulas or equations. The optimal architecture heavily relies on the unique characteristics of the dataset, including its size, complexity, and underlying patterns. Researchers need to carefully analyze the data, experiment with various architectural configurations, and evaluate the model's performance to identify the most effective architecture. While there are guidelines and best practices for neural network design, they are often based on empirical observations and heuristics rather than universally applicable mathematical formulas. Determining the optimal architecture involves a combination of domain knowledge, experience, and iterative experimentation. Researchers must adjust the architectural components, train the model, evaluate its performance, and make informed decisions based on the observed results. This trial-and-error approach is necessary to navigate the extensive design space and discover the architecture that maximizes predictive power and generalization capabilities for the specific product characteristics being predicted.

Another important limitation that has been found is that the model inconsistent ability to accurately predict whether the wafer testing phase should be removed. While it may show some promising results at times, they lack the reliability necessary to entirely eliminate the test phase. The wafer testing phase being a critical step in the semiconductor manufacturing process, where each individual dies on a wafer is tested for functionality and quality before proceeding with further processes. This phase involves a range of tests and measurements. It helps to identify any defects or issues early on, allowing for corrective actions to be taken or defective dies to be removed. While the model may exhibit reasonable accuracy in certain cases, it can also overlook subtle defects or fail to detect underlying issues that can affect the overall performance and reliability of the dies. Moreover, the model's predictions are influenced by factors such as limited training data and the inability to accurately generalize to different dies. The complexity and diversity of semiconductor manufacturing make it challenging for the model to capture all the intricacies details required for accurate predictions regarding the wafer testing phase. Given the critical importance of ensuring high-quality dies in industries, it is crucial to maintain the wafer testing phase as a vital quality control step. While the model can serve as a valuable tool for assisting in decision-making processes, it should not be solely relied upon to replace the expertise and rigorous testing procedures currently employed in the semiconductor industry.

Overcoming this limitation involves researchers to strike a careful balance between domain expertise and empirical exploration. They must have a deep understanding of the problem domain, coupled with the ability to iteratively experiment with different architectural configurations. Through this iterative process, researchers can progressively refine the architecture to enhance the model's performance and optimize its ability to predict product characteristics. It is important to recognize that the search for the optimal mathematical formula to define the architecture is an ongoing research area. Researchers continually explore new techniques, algorithms, and approaches to automate or guide the architectural selection process. Nevertheless, currently, selecting the optimal architecture for training neural networks on a given dataset remains a complex task that relies on a combination of empirical analysis, domain expertise, and experimentation. Future research questions that have emerged can be framed as follows:

— *"How can a mathematical formula be developed to determine the optimal architecture for advanced machine learning models, with the objective of achieving higher accuracy while minimizing the iteration process for architecture selection? "*

8 References

[1]. B.A. Kitchenham, D. Budgen, P. Brereton, Evidence-Based software engineering and systematic reviews, 4 CRC Press, 2015.

[2]. Faiza Allah Bukhsh, Zaharah Allah Bukhsh, Maya Daneva, “A systematic literature review on requirement prioritization techniques and their empirical evaluation”, Computer Standards & Interfaces, Volume69, 2020, 103389, ISSN0920-5489, https://doi.org/10.1016/j.csi.2019.103389.

[3]. Y. Dai and J. Huang, "A Sales Forecast Method for Products with No Historical Data," 2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), Chengdu, China, 2021, pp. 229-233, doi: 10.1109/ICCCBDA51879.2021.9442603.

[4]. V. Senthilkumar and B. V. Kumar, "A Survey On Feature Selection Method For Product Review," 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 2021, pp. 1-5, doi: 10.1109/ICAECA52838.2021.9675726.

[5]. Q. Zhou, R. Han and T. Li, "A Two-Step Dynamic Inventory Forecasting Model for Large Manufacturing," 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 2015, pp. 749-753, doi: 10.1109/ICMLA.2015.93.

[6]. K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[7]. Piyush Bhardwaj, Parul Tiwari, Kenneth Olejar, Wendy Parr, Don Kulasiri, A machine learning application in wine quality prediction, Machine Learning with Applications, Volume 8, 2022, 100261, ISSN 2666-8270, https://doi.org/10.1016/j.mlwa.2022.100261.

[8]. J. Clien, Y. Wei and Y. Zou, "Analysis of the Relevance between Title of Product and Search Term," 2022 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE), Guangzhou, China, 2022, pp. 172-176, doi: 10.1109/MLISE57402.2022.00041.

[9]. D. Friesel and O. Spinczyk, "Black-Box Models for Non-Functional Properties of AI Software Systems," 2022 IEEE/ACM 1st International Conference on AI Engineering – Software Engineering for AI (CAIN), Pittsburgh, PA, USA, 2022, pp. 170-180, doi: 10.1145/3522664.3528602.

[10]. S. A. Miraftabzadeh, P. Rad, M. Jamshidi and J. Prevost, "Customer Review Analytics using Subjective Loss Function for Conceptual-based Learning," 2018 13th Annual Conference on System of Systems Engineering (SoSE), Paris, France, 2018, pp. 211-218, doi: 10.1109/SYSOSE.2018.8428702.

[11]. S. Swarnakantha, B. Chathurika, K. V. Weragoda, W. M. I. K. Bowatte, E. V. Thalawala and M. M. U. L. Bandara, "Decision-Making Platform for SMART Plantation Agriculture Using Machine Learning and Image Processing," 2022 IEEE 7th International conference for Convergence in Technology (I2CT), Mumbai, India, 2022, pp. 1-6, doi: 10.1109/I2CT54291.2022.9825063.

[12]. H. Lu, E. Kocaguneli and B. Cukic, "Defect Prediction between Software Versions with Active Learning and Dimensionality Reduction," 2014 IEEE 25th International Symposium on Software Reliability Engineering, Naples, Italy, 2014, pp. 312-322, doi: 10.1109/ISSRE.2014.35.

[13]. H. Sharma and A. Chug, "Dynamic metrics are superior than static metrics in maintainability prediction: An empirical case study," 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), Noida, India, 2015, pp. 1-6, doi: 10.1109/ICRITO.2015.7359354.

[14]. ZhiQiang Geng, ShanShan Zhao, GuangCan Tao, YongMing Han, Early warning modeling and analysis based on analytic hierarchy process integrated extreme learning machine (AHP-ELM): Application to food safety, Food Control, Volume 78, 2017, Pages 33-42, ISSN 0956-7135, https://doi.org/10.1016/j.foodcont.2017.02.045.

[15]. F. Li et al., "Ensemble Machine Learning Systems for the Estimation of Steel Quality Control," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 2245-2252, doi: 10.1109/BigData.2018.8622583.

[16]. P. H. Chou, H. Y. Hsiao and K. N. Chiang, "Failure Life Prediction of Wafer Level Packaging using DoS with AI Technology," 2019 IEEE 69th Electronic Components and Technology Conference (ECTC), Las Vegas, NV, USA, 2019, pp. 1515-1520, doi: 10.1109/ECTC.2019.00233.

[17]. Xia Zhang, Hong Yin, Changbo Wang, Jin Wang and Yanping Zhang, "Forecast the price of chemical products with multivariate data," 2015 International Conference on Behavioral, Economic and Socio-cultural Computing (BESC), Nanjing, 2015, pp. 76-82, doi: 10.1109/BESC.2015.7365962.

[18]. C. Lork, B. Rajasekhar, Chau Yuen and N. M. Pindoriya, "How many watts: A data driven approach to aggregated residential air-conditioning load forecasting," 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, 2017, pp. 285-290, doi: 10.1109/PERCOMW.2017.7917573.

[19]. P. -Y. Du, M. Ebrahimi, N. Zhang, H. Chen, R. A. Brown and S. Samtani, "Identifying High-Impact Opioid Products and Key Sellers in Dark Net Marketplaces: An Interpretable Text Analytics Approach," 2019 IEEE International Conference on Intelligence and Security Informatics (ISI), Shenzhen, China, 2019, pp. 110-115, doi: 10.1109/ISI.2019.8823196.

[20]. K. C. -C. Cheng et al., "Machine Learning-Based Detection Method for Wafer Test Induced Defects," in IEEE Transactions on Semiconductor Manufacturing, vol. 34, no. 2, pp. 161-167, May 2021, doi: 10.1109/TSM.2021.3065405.

[21]. A. Jauhri, B. McDanel and C. Connor, "Outlier detection for large scale manufacturing processes," 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA, 2015, pp. 2771-2774, doi: 10.1109/BigData.2015.7364079.

[22]. Y. Jin, J. Yin, H. Zhang, J. Zhang, Z. Zhang and D. Yang, "Parameters Predictions of Paddy Grain Drying Based on Machine Learning," 2021 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Xi'an, China, 2021, pp. 701-707, doi: 10.1109/ICITBS53129.2021.00176.

[23]. Y. I. Eremenko, D. A. Poleshchenko and Y. A. Tsygankov, "Prediction of Quality Indicators of Iron Ore Processing Operations Using Deep Neural Networks," 2020 2nd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), Lipetsk, Russia, 2020, pp. 425-429, doi: 10.1109/SUMMA50634.2020.9280676.

[24]. C. Huang, B. W. -K. Ling and X. Ding, "Relationship between statistics and filters in noninvasive blood glucose estimation analysis," 2022 IEEE International Symposium on Product Compliance Engineering - Asia (ISPCE-ASIA), Guangzhou, Guangdong Province, China, 2022, pp. 1-4, doi: 10.1109/ISPCE-ASIA57917.2022.9970906.

[25]. J. Tian, Y. Wu, S. Liu and P. Liu, "Residential load disaggregation based on resident behavior learning and neural networks," 2017 IEEE Conference on Energy Internet and Energy System Integration (EI2), Beijing, China, 2017, pp. 1-5, doi: 10.1109/EI2.2017.8245665.

[26]. T. Zhang, Q. Du, J. Xu, J. Li and X. Li, "Software Defect Prediction and Localization with Attention-Based Models and Ensemble Learning," 2020 27th Asia-Pacific Software Engineering Conference (APSEC), Singapore, Singapore, 2020, pp. 81-90, doi: 10.1109/APSEC51365.2020.00016.

[27]. Sebastian Schorr, Matthias Möller, Jörg Heib, Dirk Bähre,Quality Prediction of Drilled and Reamed Bores Based on Torque Measurements and the Machine Learning Method of Random Forest, Procedia Manufacturing,Volume 48,2020,Pages 894-901,ISSN 2351-9789,https://doi.org/10.1016/j.promfg.2020.05.127.

[28]. T. Roosefert Mohan, J. Preetha Roselyn, R. Annie Uthra, D. Devaraj, K. Umachandran, Intelligent machine learning based total productive maintenance approach for achieving zero downtime in industrial machinery,Computers & Industrial Engineering,Volume 157,2021,107267,ISSN 0360-8352, https://doi.org/10.1016/j.cie.2021.107267.

[29]. Alrufaihi D., Oleghe O., Almanei M., Jagtap S., Salonitis K, Feature Reduction and Selection for Use in Machine Learning for Manufacturing (2022) Advances in Transdisciplinary Engineering, 25, pp. 289-296.

[30]. Fatih Yucalar, Akin Ozcift, Emin Borandag, Deniz Kilinc,Multiple-classifiers in software quality engineering: Combining predictors to improve software fault prediction ability, Engineering Science and Technology, an International Journal, Volume 23, Issue 4, 2020,Pages 938-950,ISSN 2215-0986, https://doi.org/10.1016/j.jestch.2019.10.005.

[31]. Benjamin Maschler, Sophia Tatiyosyan, Michael Weyrich,Regularization-based Continual Learning for Fault Prediction in Lithium-Ion Batteries,Procedia CIRP,Volume 112,2022,Pages 513-518,ISSN 2212-8271, https://doi.org/10.1016/j.procir.2022.09.091.

[32]. Singh, S., Singla, R. Defect prediction model of static code features for cross-company and cross-project software. Int. j. inf. tecnol. 13, 667–675 (2021). https://doi.org/10.1007/s41870-018-0262-5.

[33]. Zhang, H., He, X., Yan, W. et al. A machine learning-based approach for product maintenance prediction with reliability information conversion. Auton. Intell. Syst. 2, 15 (2022). https://doi.org/10.1007/s43684-022-00033-3.

[34]. Iliadis, D., De Baets, B. & Waegeman, W. Multi-target prediction for dummies using two-branch neural networks. Mach Learn 111, 651–684 (2022). https://doi.org/10.1007/s10994-021-06104-5.

[35]. Fu, H., Liu, Y. A deep learning-based approach for electrical equipment remaining useful life prediction. Auton. Intell. Syst. 2, 16 (2022). https://doi.org/10.1007/s43684-022-00034-2

[36]. Li, W., Zhang, L., Chen, S. (2019). Attention Network for Product Characteristics Prediction Based on Reviews. In: Gedeon, T., Wong, K., Lee, M. (eds) Neural Information Processing. ICONIP 2019. Communications in Computer and Information Science, vol 1143. Springer, Cham. https://doi.org/10.1007/978-3-030-36802-9_21

[37]. Grand View Research. (2021). Automotive Semiconductor Market Size, Share & Trends Analysis Report By Component (Memory, Logic ICs), By Application (Powertrain, Safety Systems), By Region (APAC, North America), And Segment Forecasts, 2021-2028. Retrieved from https://www.grandviewresearch.com/industry-analysis/automotive-semiconductor-market.

[38]. NOVEL MACHINE-LEARNING BASED IC TESTING STRATEGY, UNIVERSITY OF BOLOGNA, Fabrizio Finelli

[39]. Peffers, Ken & Tuunanen, Tuure & Rothenberger, Marcus & Chatterjee, S.. (2007). A design science research methodology for information systems research. Journal of Management Information Systems. 24. 45-77.

[40]. Wieringa RJ. Design science methodology for information systems and software engineering. London: Springer, 2014. 332 p. doi: 10.1007/978-3-662-43839-8

[41]. A. Verma, C. Kapoor, A. Sharma and B. Mishra, "Web Application Implementation with Machine Learning," 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2021, pp. 423-428, doi: 10.1109/ICIEM51511.2021.9445368.