Machine Learning-based Predictive Analytics in ERP Systems

Information and Communication Technology/Faculty of Technology
Master's thesis in technology

Author:
Rasmus Jokinen

21.5.2026
Turku

Master's thesis in technology

Subject: Software Engineering
Program: Information and Communication Technology
Author: Rasmus Jokinen
Title: Machine Learning-based Predictive Analytics in ERP Systems
Supervisors: Ville Leppänen, Mika Murtojärvi
Number of pages: 54 pages
Date: 21.5.2026

This thesis studies applying machine learning-based predictive analysis in ERP systems. A literature review is conducted to see what modules and features have had these integrations in ERP systems before, as well as what kinds of advantages they could bring for a company. In the case study of the thesis, a machine learning model is trained for the project management module for predicting project budgets.

The thesis shows that machine learning-based predictive analysis is most used in the inventory management module and that it can provide a lot of benefits, mainly by optimizing processes. The trained machine learning model did not achieve an accuracy high enough for an integration to the system but showed potential in ML-based project budgeting in ERPs. Further efforts should be directed to increasing the size of the dataset.

Key words: Enterprise Resource Planning, machine learning, predictive analysis

Table of contents

1 Introduction

1.1 Background

Enterprise Resource Planning system (ERP) is a software solution that integrates several modules, each built to handle a specific business function [1]. This way an ERP forms a unified interface for managing core business processes [2].

Organizations can gain an advantage in the market by utilizing an ERP. It enhances the effectiveness of management while lowering the error rate of decision-making [3]. An ERP saves time and lowers costs in organizations out of the box, as it eliminates the need for separate software applications for each business process and the need to manually input data into each of them separately [1]. Therefore, by integrating all or most business processes into the same system, organizations can gain new possibilities as the data can be automatically shared between modules. Comparing utilization of an ERP to a situation where each department of a company operates in their own systems, an ERP saves time and effort by reducing bureaucracy. On the other hand, deployment of an ERP can be hindered by the distrust of high-tech solutions by organization's personnel and the possibly long and expensive process [4].

Integrating many business processes as modules into a system and then sharing all the data they generate between the modules is exactly why machine learning (ML) is interesting in ERPs.

1.2 Machine Learning

Machine learning means that a computer adapts its actions, making them more accurate. The accuracy is measured by comparing the computer's actions to the correct ones, which creates an element of learning to the process. [5]

With ML, the large amount of data gathered in an ERP can be analyzed to provide organizations with better, data-driven insights [1]. This means that companies can use ML algorithms to analyze their own business data to ultimately help, for example, with decision-making and automation. ERP providers like Oracle have already been building ML solutions into their products to aid with many functionalities [1].

One of the applications for machine learning can in ERPs is predictive analytics.

1.3 Predictive Analytics

In ML-based predictive analytics the ML algorithms forecast probable outcomes using historical data [1]. There are many applications for predictive analytics in ERP systems as there are many integrated parts and lots of data to forecast from. For example, organizations can improve storage levels, anticipate customer needs or optimize the supply chain [1].

ML-based predictive analytics gives organizations the possibility to make strategic decisions based on the predictions that the ML algorithms and models make. This way, predictive analysis might uncover patterns and associations that were left unnoticed as they are difficult to reveal without thorough analysis. All in all, an ERP system with integrated predictive analytics will improve the competitiveness of an organization. [6]

Interestingly, adoption of predictive analytics in ERP systems has been outpaced by other AI integrations. Adoption of AI often starts with chatbots and automation of processes because they have a lower barrier of entry for organizations. Predictive analytics requires a proper database and understanding of the data, which is not often achieved in the beginning of ERP system setup. [7]

Further issues with ML-based predictive analytics include data-accuracy problems and the usage of complex ML algorithms [1].

1.4 Goal

This research is conducted in collaboration with a small to mid-sized company that is developing its own ERP system. The company has previously used a third-party ERP system targeted for the industry, but the system has many issues and does not fit the company's needs perfectly. This has driven the company to choose to implement its own system instead.

Developing an in-house ERP system gives the company freedom to include all the features it deems necessary and beneficial. Integrating AI, namely predictive analytics, into the new ERP system was quickly deemed to have a lot of potential. Even though the previous system lacked artificial intelligence, the data of about two years of using it can be utilized in training predictive machine learning models for the new ERP.

The goal of this study is, firstly, to find potential use-cases for ML-based predictive analysis in the new ERP. This means that the study should propose modules and features where ML-based predictive analysis could be integrated into. The secondary goal of the study is to

uncover how much potential ML-based predictive analysis can have in the new system. The company needs to know what benefits it gets from this AI integration. The results will not only show whether the benefits are great enough to go forward with a larger implementation in the future but also help prioritize modules and features where the integration is most needed. This means that the study should not only provide applicable examples, but measurable results as well.

The goals are summarized as the following research questions:

- RQ1: In which functionalities of an ERP system can machine learning-based predictive analytics be utilized?
- RQ2: What are the benefits of using machine learning-based predictive analytics in ERP systems?

1.5 Methodology

The research process will follow design science research. The goal of the study demands that there is both general knowledge gained from the study, but also actionable results for the company itself. Therefore, this study has two parts with two distinct research methodologies to answer the research questions.

Firstly, a literature review is conducted to answer the research questions generally to ensure that there is clear direction for adopting machine learning-based predictive analytics in the company moving forward. The literature review aims to gain knowledge to determine where and why ML-based predictive analytics should be used in the company. Secondly, a case study is conducted to answer the second research question by providing measurable results for the company in an actual attempt in training an ML-model for predictive analysis. The results from the case study can then be used to evaluate whether ML-based predictive analytics is currently beneficial in the company's system.

1.6 Structure

The research is divided into a literature review and a case study. First, the literature review is conducted. The ERP systems and their usual modules are explained in Chapter 2. Then the information gained from this is utilized in Chapter 3, which explains which modules and functionalities in ERPs can benefit from ML-based predictive analytics.

The case study is introduced and executed in Chapter 4. After this, in Chapter 5, results from both the literature review and the case study are discussed in depth and finally the conclusions and summarized answers to the research questions are provided in Chapter 6.

2 Enterprise Resource Planning (ERP) Modules

Enterprise Resource Planning (ERP) systems are divided into modules, and each module is focused on its own business function. Therefore, companies often select a subset of the modules that an ERP provider has to offer [9]. Similarly, not all the modules are explained in this chapter as the focus is mostly on the modules that are critical in the ERP system that is the setting of the case study in Chapter 4. The critical modules include human resource management, inventory management, and project management modules. In addition, finance and sales modules are one of the most used modules in ERPs [9]. They will be visited as well. In this chapter, the modules are gone through one by one to fully understand them and the functionalities they have. Chapter 3 goes into detail about how ML-based predictive analytics can be implemented within these modules.

Sometimes each individual piece of an ERP is called a module despite its size. For example, the ERP provider Odoo has a list of all the ERP's functionalities, which includes 47 modules divided into 9 categories [8]. This study refers to the general categories as modules, and the modules inside the categories as sub-modules. The reason for this terminology is simplicity. There is little benefit to go through each individual sub-module as they are highly ERP provider specific. On the contrary, the modules are about the same in all ERP systems.

2.1 Human Resource Management Module

Human Resource Management (HRM) module includes all the same features that are available in a workforce management application [9]. Generally, an HRM module stores employee information and documents like performance reviews and job descriptions [10]. In addition, the module offers a variety of functionalities like time tracking, absence management, and employee benefits management [9].

The benefits of using an HRM module in an ERP system are straightforward. As the HRM stores a lot of data on all employees, it works as a single source of truth to all other modules that need employee data. HRM modules are essential to many organizations to eliminate duplicate or inaccurate data which, without an HRM module, would otherwise be stored in many different places around the system [9]. The module also provides managers with better tools to allocate their team's time and resources which helps create a more productive and profitable business [10].

SAP [11] and Oracle [12] both list features of their ERP system's HRM module on their websites. On paper, Oracle's HRM module is a self-sufficient Human Capital Management (HCM) software, but it is possible to integrate it seamlessly into the Oracle's ERP, so it can be treated as an ERP module as well. Both ERP providers name similar functionalities in their HRM modules: time tracking, absence management, employee benefits, payroll functionalities, reports and dashboards. They also mention that they have a single data source for all HR information. In addition, both have a great amount of localization built into the module and SAP's module description mentions that the system gets automatic updates for legal and regulatory changes.

SAP and Oracle both have a sub-module called Talent Management which focuses on career development. It offers tools to source and recruit possible candidates, onboarding new employees and training experienced ones. The only glaring difference between the two ERP providers is that SAP seems to have invested more in the document management side of the HRM module as it provides automated document generation with document templates. However, it is important to note that both ERP providers in question can have left out some details of the module's functionality. All in all, the collection of features is surprisingly similar.

2.2 Inventory Management Module

Inventory management is significant for financial performance of a business. In its core, managing inventory is a constant balance between reducing costs of excessive storage but keeping enough inventory to serve the customers' and the company's needs [13]. Reducing inventory is one of the major reasons for companies to adopt an ERP [14] and it is also one of the goals of the inventory management module.

ERP providers often provide a warehouse management module in addition to the inventory management module. These two have different purposes. Inventory management lets the organization track its inventory quantities in real time and manages the transaction history [15], whereas warehouse management creates effective warehouse processes like directing picking new orders and guiding incoming deliveries [16].

The inventory management module provides functionality to set reorder points and calculate safety stock. These can be computed based on the demand history, which the module tracks.

The module also provides dashboards and analytics to give out recommendations based on the history of demand. [17]

The module has many potential benefits for companies that integrate it into their workflow. Effective inventory management in manufacturing will lead to continuous supply of materials for production. It decreases delays in production schedules and lowers inventory costs but also causes less delay in deliveries and can increase customer satisfaction. [15] However, errors in the module's provided functionalities can result in stock-outs or high stock, both of which are financially costly to the company [13]. Moreover, it is possible for the physical quantity of materials in the inventory to not match with the quantity calculated in the system, which might be caused by human error [15].

2.3 Project Management Module

It is essential to have a good project management strategy built into an organization for it to carry out successfully accomplished projects as up to 75% of projects can fail solely because of project management issues [18]. The project management modules in ERPs aim to minify these issues as ERP solutions improve project management [19]. A project management system can be a stand-alone application or a module in a larger ERP system [18].

Traditionally, a project management module includes scheduling, planning, and tracking a project [18]. It shows the project progress, cost, and resources in real time [19]. It should allow specifying critical deadlines, activities and schedules. Monitoring and coordinating project resources is also an important feature, for both material and human resources. Result analysis and improvement insights should be given when a project is closed. [20] Other features can include for example work time optimization, task prioritization, generating update reports for stakeholders, creating offers for new projects, detailed reports for operational and financial state, and a dashboard for viewing all the company's project [18].

The downside of the project management module among other modules in a popular ERP system is that it is essentially built to fit the common best practices of the industry that the system is built for. Some companies use their own practices that differ from those that the ERP is built upon. This means that the company must either heavily customize the ERP themselves or change their own practices to fit that of the ERP. [19]

2.4 Sales Module

The sales module is often also called the sales and distribution module. It supports the many activities around selling and delivering products to customers. It aims to simplify the process of selling based on customer needs. It also creates informational reports about sales and deliveries, keeps data on the customer base, determines appropriate services for customers, and forecasts product demand. [21]

The sales module operates with great amount of informational data. For this reason, it integrates with many other modules that can leverage the data as well. The module can be integrated with production schedules and inventory management to improve inventory levels and production lines and share information across an organization for decision making [22].

From the sales module's features, sales forecasting has been researched considerably. It is a feature where the ERP's data-driven architecture gets to be used heavily. Sales forecasting means estimating the demand of a product during a timeframe [21]. The traditional sales forecasting methods do not work well with complex business data and are also constantly getting less effective as the requirements of an organization increase [23]. This means that the traditional methods give less accurate forecasts over time when the organization evolves.

Sales forecasting enables an organization to use information about customer requirements and warehouse capabilities in the production of goods. It can be used in planning; purchasing, budgeting, and determining the required amount of workforce can be done with the aid of sales forecasting. It integrates with inventory management to have less risk of understock and overstock. Forecasting sales is, however, not a simple task as it can be affected by the global economy, demand for products, and the surrounding business competition. [21]

The benefits of a successful sales module are great, but the drawbacks must be also noted. The sales module supports activities that are not as structured as the activities in other modules and is therefore more difficult to implement correctly [24]. This means that some companies might choose to implement the sales module only after simpler modules are developed first or they might leave it unimplemented altogether.

2.5 Financial Module

An ERP financial module, like the other modules presented in this literature review, benefits from sharing data between modules. By integrating the module into other enterprise functions,

the financial management system is enriched with multiple viewpoints of the business. This lowers the risks in financial control and can improve the overall development of the company [25]. Some literature differentiates between a financial management module and a financial control module. In this case, the financial management module is mostly responsible for measuring the company's financial performance while the control module is used to perform tasks related to the company's finances [20]. In this section, both modules will be discussed and their functionalities introduced.

Like many other modules, their key benefits come from centralizing data to the ERP system for other modules to use. The financial module is not an exception as the module integrates and manages various financial data [25]. All financial documents are stored by the module [20]. Simply the data in the context of a larger ERP system provides possibilities. Other departments can efficiently use financial information available via the module which prevents delays that would otherwise happen [25]. In the financial module itself, the core data is used to prepare financial statements [25] and is helpful during financial audits [20]. Fixed assets are not to be forgotten either. The financial module keeps count of them and helps with managing them as well [25].

For data to be useful, it must be input to the system first. The module is suited for data input and automating some of the tasks as well. For example, registering vouchers for the general ledger and processing invoices is done in the module [25]. The financial module can also automatically create invoices from the organization's billable tasks [26]. Customer orders can also automatically generate vouchers for the company's general ledger [25].

The module includes lots of features to perform finance-related tasks. Creating and sending invoices, as well as payments can be handled within the module [26]. Another key task is to monitor and control finances around the enterprise. Monitoring and planning payments of the company, controlling the use of funds in each department, and controlling costs and profits across the business are supported [20]. The module can also automatically utilize the financial data to help with the tasks. For example, the module can give a warning when the set recovery period for accounts receivable has been exceeded, or the payroll features can automatically account for employees' salaries and other related expenses [25].

2.6 Feature Summary

This chapter went through five modules that were determined critical for the company's ERP system. Table 2-1 shows the summary of features found in these modules. ML-based predictive analysis for the five modules is gone through in Chapter 3, where Table 2-1 works as a reference point.

Table 2-1. Summary of ERP features

Module	Feature
All modules	F0.1: Dashboards F0.2: Reports
Human resource management module	F1.1: Employee information storage F1.2: Time tracking F1.3: Absence management F1.4: Employee benefits management F1.5: Payroll F1.6: Talent management
Inventory management module	F2.1: Reorder points F2.2: Safety stock evaluation F2.3: Demand history tracking
Project management module	F3.1: Scheduling, tracking, and planning projects F3.2: Project resource management F3.3: Improvement insights F3.4: Work time optimization F3.5: Task prioritization F3.6: Project offer creation
Sales module	F4.1: Customer base data storage F4.2: Sales forecasting (product demand) F4.3: Customer need determination F4.4: Data integration into other modules
Financial module	F5.1: Financial document storage F5.2: Financial statement creation F5.3: Fixed asset management F5.4: Invoice processing F5.5: Automatic and manual invoice creation F5.6: Invoice sending F5.7: Payment handling F5.8: Management of funds F5.9: Management of costs F5.10: Accounts receivable warnings F5.11: Data integration to payroll

3 Machine Learning-based Predictive Analysis in ERP Systems

In addition to a case study conducted later in Chapter 4, a literature review is done to answer both RQ1 and RQ2. This chapter will carry out said literature review. The goal is to find existing applications of machine learning-based predictive analysis in ERPs from scientific research. Answering the research questions based on a literature review will provide the company with more ideas for the future than a single case study could. As an added benefit, the knowledge gained from it can be utilized in conducting the case study, if applicable.

This chapter will first explain what methods were used to select the reviewed literature. Then, all the selected research articles will be gone through one by one in the context of this thesis. The articles will be compared to Table 2-1 to categorize the findings into modules and features that were covered in Chapter 2. Lastly, the results of the whole literature review will be discussed and summarized.

3.1 Methodology

In this thesis, the selection of relevant articles was executed in four steps. The steps are composing the search terms, determining how to limit and sort the articles, deciding the exclusion criteria for the articles, and finally conducting the selection in practice. These steps are gone through in the introduced order, and intermediate results collected during the search will be presented as well.

The research articles were searched from three research databases: ACM Digital Library, IEEE Xplore and Web of Science. The search was done during October 2025.

The search query had to accurately capture multiple aspects of the thesis' topic: predictive analysis, machine learning and ERP systems. However, some articles might describe predictive analysis with a different term or leave the solution uncategorized, never mentioning predictive analysis in the first place. The same goes for machine learning, which could be only categorized as artificial intelligence. Strictly searching for articles that use both terms in the text can potentially miss perfectly relevant and interesting articles. It was decided to use an "OR" Boolean operator between these terms to include articles that might refer to one of the terms differently. ERP was deemed too crucial of a term to be chained with the other terms this way. This decision might lead to missing relevant articles that are not strictly made for ERPs but other similar systems. Either way, the context of enterprise resource planning systems is too central for the thesis for the keyword to be left out.

With the former reasoning in mind, the following search query was formed: *(machine learning OR ml OR predictive analysis OR predictive analytics) AND (erp OR enterprise resource planning)*.

The articles found with the search query were then limited to only include articles published in 2015 at the earliest. The assumption was that most of the articles found are from the last couple of years as AI in ERPs has been gaining traction during this time. However, older articles might be as relevant as the more recent ones, and so it was decided to include articles from the last 10 years. Table 3-1 shows the number of articles found with these settings and the formerly introduced search query.

For finding the most relevant articles to be included in the review, the articles were naturally sorted by relevance. Only the first 100 articles per database were considered to speed up the process and to eliminate unnecessary work in preparation for a large number of results from the databases.

Table 3-1. Number of initial search results in research databases

Database	Initial number of articles
ACM Digital Library	2157
IEEE Xplore	984
Web Of Science	878

Research articles were excluded from selection based on multiple criteria. The goal was to only have to look through the search results once, going through all criteria for each research article right away. Some articles could be left out rather quickly, but others needed a closer look into the text to properly categorize the methods used in the research. First, the article was ruled out immediately if the title or the abstract did not include anything related to the topic. Then, if the article was mainly a literature review or did not propose a practical solution by means of a case study for example, it was left out. Lastly, the article was excluded if the artificial intelligence used in the paper could not be categorized as machine learning, the machine learning-based solution could not be categorized as predictive analysis, or the article did not mention ERPs at all.

Table 3-2 includes the number of articles included in the literature review after going through 100 articles in each chosen database and comparing the above exclusion criteria to each of them. A total of 18 articles were found. However, during the writing process of the literature

review, one more paper was left out as it was a classification task, and the solution could not be perceived as predictive analytics. This decreased the number of articles to 17. The chosen articles are listed in Table 3-3.

Table 3-2. Number of chosen articles from each research database

Database	Number of articles chosen
ACM Digital Library	1
IEEE Xplore	9
Web Of Science	7
Total	17

As hypothesized before conducting the search of articles, most papers were recently published. The search was limited to papers published in 2015 at the earliest, in fear of missing relevant older literature. Of the 17 chosen articles, 9 were published in the last two years and no paper of the set of included articles was published earlier than 2019. It is safe to say that the survey will focus on the more recent advancement in this field.

All articles are gone through one by one in the following survey. Each research paper will be reviewed and its most important findings and results highlighted. All articles are discussed together in Section 3.3.

Table 3-3. Articles included in the literature review

Title	Database	Year
[3] Machine Learning-based Predictive Analytics for Financial Planning and Budgeting in ERP systems	IEEE Xplore	2024
[23] The Design of ERP Intelligent Sales Management System	Web Of Science	2020
[27] Demand Forecasting Based on Machine Learning for Mass Customization in Smart Manufacturing	ACM Digital Library	2019
[28] AI-Powered Smart Inventory Management: Enhancing Efficiency Through Predictive Analytics and Automation	IEEE Xplore	2025
[29] Advanced Predictive Maintenance with Machine Learning Failure Estimation in Industrial Packaging Robots	IEEE Xplore	2020
[30] Predictive Analysis Methodology for Industrial Systems: Application in Supplier Delays Prediction	IEEE Xplore	2022
[31] System Design for a Data-Driven and Explainable Customer Sentiment Monitor Using IoT and Enterprise Data	IEEE Xplore	2021
[32] Deep Learning-Based Optimization of Cloud Enterprise Resource Planning (ERP) Systems for Adaptive Decision Support and Management Effectiveness Analysis	IEEE Xplore	2024
[33] Integrating Machine Learning Based Sales Forecasting with Odoo Erp for Automated Inventory Management in a Retail Company	IEEE Xplore	2025
[34] Application of Machine Learning Methods for Production Data Analysis	IEEE Xplore	2025
[35] Prediction of Cloth Waste Using Machine Learning Methods in the Textile Industry	IEEE Xplore	2022
[36] A proposed electric/hybrid batteries recycling hub with ERP implementation and simulation	Web of Science	2025
[37] "A comprehensive review of AI-enhanced decision making: An empirical analysis for optimizing medication market business"	Web of Science	2025
[38] Forecasting and Inventory Planning: An Empirical Investigation of Classical and Machine Learning Approaches for Svanehoj's Future Software Consolidation	Web of Science	2023
[39] Machine Learning-integrated digital twins for process optimization in Industry 5.0	Web of Science	2025
[40] Automated machine learning for fabric quality prediction: a comparative analysis	Web of Science	2024
[41] Prototyping Machine-Learning-Supported Lead Time Prediction Using AutoML	Web of Science	2021

3.2 Analysis

This analysis goes through all previously chosen research articles that include machine learning-based predictive analysis in ERP systems. Table 3-3 includes all the articles. They are analyzed one by one, from top to bottom. The articles are analyzed from the perspective of the research questions.

For RQ1, the module and functionality the research article concentrates on is uncovered. For this reason, the findings are also attempted to be mapped to Table 2-1, which includes the found functionalities of the five modules focused on in Chapter 2. As the literature review did not include an exclusion criterion that the module centered upon in the articles should be one of the modules from Chapter 2, this might not always be possible. For RQ2, the benefits of machine learning-based predictive analytics gathered from the articles are gone through. More precisely, this means the benefit the company would receive from integrating the proposed or implemented solution. If a study does not provide such metrics from an example company, the benefits might be theorized upon. If not, the second research question will be left unanswered for the study in question.

In addition to analyzing the articles based on the research questions of the thesis, technical details about the machine learning models presented in the articles are pointed out. This is done for the case study which is conducted in Chapter 4. There are a lot of points of interest in training a successful machine learning model for an ERP system. For example, the machine learning method used, the location, quality and quantity of data used to train the model, the programming languages, tools and software libraries used in implementing the model and so on. Even if these findings are eventually not used in the case study of the thesis, they provide valuable information for the company when it chooses to move forward with a larger integration of machine learning in the company's ERP system.

In article [3], a framework for a financial module of an ERP system is proposed. The framework is divided into multiple components, one of which is for data-driven decision-making. For this proposed component, a machine learning model for guiding the budgeting of an enterprise is trained. This article, with budgeting as its focus, can therefore be mapped into two features of Table 2-1, which are F5.8: management of funds and F5.9: management of costs. A multitude of different approaches are evaluated but a support vector machine (SVM) based on Particle Swarm Optimization (PSO) i.e. SVM-PSO performs better than the other methods. It is not clear from the article how the training data was gathered, what the data was

and how much of it there was. This makes it hard to evaluate the importance of the findings of this article and how they relate to ERP systems on a practical level. The results are not that usable, but it can be determined that using ML-based predictive analytics to guide budgeting decisions is at least present in the research literature.

In [23], the authors design a system to predict future sales. The article is clearly about the sales module in ERPs and more precisely about the feature F4.2: sales forecasting. They train a backpropagating neural network on historical sales data to make predictions. The model is used with multiple products. The goal of the system was to mine potential patterns from the data and predict customer behaviour, so that the decision makers in the company can adjust market strategies and reduce risks. The ERP system utilizes the results from the deep learning model in many ways: finding potential customers, pushing advertisements to customers and allocating the company's resources based on the predicted sales numbers. After developing the system, the authors did an experiment using the old system in parallel with the new one. During a six-month period, the number of products sold increased when using the predictive system: from an average of 166 to 211 products sold per month. They made the following conclusions about the results. Firstly, customer satisfaction was increased as the customer needs were better understood, which led to proper services being provided to them faster. Secondly, inventory management was improved because the predictions improved production planning. Inventory backlog and safety stock were reduced because of the improved inventory management.

Like [23], the authors of [27] have also based their study on the feature F4.2. They train a Long short-term memory (LSTM) neural network for forecasting product demand. The model is not integrated into an ERP system directly, but the article mentions that it could be integrated into an inventory management module. The model is trained on demand data of four years and is evaluated by comparing its predictions to the data of the fifth year. As the model is not integrated into an ERP system, it is likely that the training data is not from one either, but the origin of the data is not disclosed in the article. The trained model is compared to a more traditional time series model ARIMA. In the initial stage, ARIMA performed better than the neural network, but the long-term performance of the LSTM model was better. Overall, the LSTM was 19,1% more performant based on error rate calculated with root mean square error. Interestingly, the demand forecast is later used in calculating the optimal stock levels of the manufacturer's products. The article does not present any further benefits the

model might have when integrated to a company but as it is about sales forecasting much like article [23], it can be assumed that the same benefits would apply here as well.

The article [28] presents an entire smart, AI-driven inventory management system. In addition to using predictive analytics, the article talks about automating tasks and implementing security measures into the system. Predictive analytics are used for sales forecasting. The numbers from that are used to determine stock levels. This would mean that not only does the article centre on the inventory management module, but it also has functionality of the sales module built in, at least when referring to the Table 2-1. The features of these models are F2.2: Safety stock evaluation and F4.2: Sales forecasting. However, it seems that the sales forecast is done specifically for optimizing stock levels, so it leans more on the side of F2.2.

The authors use historical sales data from six months to predict the demand for the next days or weeks. The data was gathered from two sources. Firstly, from actual sales records, and secondly from the smart inventory management system as it tracks all outgoing packages by automatically scanning the QR codes of all incoming and outgoing packages. They used Python libraries, i.e. Pandas and NumPy, for pre-processing the data and eventually to train the models. Eventually, the model was trained on an 80%/20% split into training and testing data. Random Forest and Linear Regression were used as the machine learning models, from which, Random Forest performed better. They evaluated both models with mean absolute error, root mean squared error and the R square value and with these evaluation methods, the Random Forest model was more accurate and had lower error rates. After the system was in place, the authors ran a one-month comparison test to the old manual system. The inventory accuracy improved by 4,4%. The stockouts were reduced by 25% and overstocking incidents by 18%. Overall, the inventory holding costs were 15% less than with the manual system, showing that companies can save on inventory costs by applying machine learning to the inventory management module. The results are not completely comparable to the topic of the thesis as the article created a complete smart inventory management system instead of just integrating predictive analytics, but the results can still give an indication of what type of benefits the predictive analytics could have had alone.

In [29], a machine learning model is trained to predict production machinery failures. The machinery is overseen by an ERP system and the training data for the model is gathered only from the ERP system itself and not from IoT devices for example. The article does not state details about the ERP system used, nor what module the ML model integrates into, however,

the module would most likely be a manufacturing execution system (MES) which can be integrated into an ERP as a module, as explained by SAP, an ERP provider [42]. SAP even mentions maintenance management as one of MES's features. For training the model, the authors used data from the last 3 years. It included downtime records, the machine working hours between failures, the number of products produced, and the product that was produced when the failure happened. In total, this amounted to 157 datapoints, from which 146 were used for training the model and the remaining 11 to test it. The model used was a multilayer perceptron (MLP) which had 8 inputs, 20 hidden layers and 4 outputs. The outputs were failure information: the workshop, machine, failure area and operation time. What the authors ultimately tested though, was only the operation time until next failure. The model achieved an accuracy of 91%. No practical tests were done with the model, but it is hypothesized that downtime costs will be greatly reduced as the predictions make it possible to schedule maintenance for machinery before the actual failure will occur.

Article [30] proposes a methodology for improving the quality of industrial data. This is interesting in the scope of the case study done in this thesis, but this would not be relevant to the literature review by itself. Luckily, the authors also make a case study for their methodology, in which they predict supplier delays with a machine learning approach. Like the article [29] before, this study is not about one of the five core modules that are included in Table 2-1. The topic of this article would most likely fall under the supply chain management module, or more precisely its procurement sub-module which is responsible for securing materials for manufacturing [9]. All the data used for the predictions is collected from the ERP system of the study's partner company. The data includes over 20000 datapoints and each of them contains the average delivery delay of a week based on the supplier and the product delivered. The data contains over 100 suppliers and nearly 4000 products. Three machine learning algorithms are used: decision tree, random forest, and naive Bayes. Considering accuracy, precision, recall, and F1-score, the random forest model was evaluated to perform the best out of the three models with an accuracy of 76,02%. The naive Bayes model was deemed unsuitable, and it performed the worst. The paper does not discuss the benefits the partner company would receive by integrating this predictive ML-model into their workflow.

The authors of [31] implement a machine learning model that predicts customer escalations utilizing IoT and enterprise data. This is done from the perspective of manufacturing companies that have to monitor the performance of an installed system to keep customer

satisfaction high. As stated in the article, the enterprise data used in training the model is often found in a Customer Relationship Management (CRM) module of an ERP. Therefore, this study falls out of the set of modules discussed in Chapter 2. The data used is two-fold. They use log data gathered from the installed systems and enterprise data gathered from an ERP system. The enterprise data includes customer service tickets about any interaction or problem, spare parts used for maintenance or repair, and customer contracts. They use one deep learning method and one decision tree method for the task. The deep learning method is a long short-term memory, and the decision tree is an ensemble that includes Random Forest and XGBoost. The authors solely used recall for evaluating the models and found that using just a Random Forest method provided the best results with a recall of 42,46%. The authors estimate that a 5-person customer support team could prevent close to 49% of customer escalations per year. Furthermore, using the implemented system can be useful in understanding what features lead to escalations more often than others. The model can therefore help improve customer satisfaction and product management as the company's efforts can be focused on the most important issues.

In their study, authors of [32] attempt to largely integrate deep learning into an ERP system for optimizing the system as a whole. They implement three machine learning models, each implemented for a specific task. The interesting model from the point of view of this thesis is the two-layer LSTM model trained for predicting future sales and inventory demand. The authors state that this machine learning model is integrated into the inventory management and finance modules. Interestingly, the feature of sales forecasting (F4.2) falls under the sales module in Table 2-1. The study uses experimental data gathered from public datasets as well as partner companies. Because of the large scope of the different models, the data includes a lot of different features and has close to 1,2 million data points. For preprocessing data, they use Pandas and Scikit-learn libraries, and they use the frameworks TensorFlow and PyTorch for training the model. They evaluated the model based on its accuracy and its key business metrics: response time and resource utilization. In other words, they wanted to know how accurately, in which amount of time and with what resources the model can perform. The model reached an accuracy of 95% with a response time of 0.5 seconds. The article states that the model uses 65% of the CPU and 60% of RAM, but having not disclosed the machinery running the model, the resource utilization statistics remain rather meaningless. As the authors have focused on a more all-around integration of deep learning into ERP systems, the benefits

are also presented with that in mind. However, the sales forecasting model resulted in an improvement of 25% in inventory turnover.

In [33], a machine learning-based sales forecasting model is trained and integrated into Odoo, an open-source ERP provider. The solution is implemented as a custom module inside the ERP and integrated directly into the inventory management module so that the predictions update the processes in real time, automating inventory management related decisions. Again, like for articles [27], [27], [28], and [32], sales forecasting (F4.2) falls under the sales module. This time, it is also clearly integrated into the inventory management module, which was not directly done in the previous articles. The training data was gathered from an ERP system of a partner company, and the dataset includes historical sales data from January 2024 to January 2025. The authors chose XGBoost as the machine learning method as it had been proven to be effective in earlier research literature. The model had a mean squared error of 0,78 and root mean squared error of 0,89. To evaluate the model on key performance indicators of the business, a simulation was created. In the simulation the inventory turnover improved significantly by 230% and the total operational costs reduced by 8,5%. The authors also concluded that automating inventory replenishment with this kind of predictive analytics approach can reduce holding costs, improve customer satisfaction, and lead to a substantial increase in operational efficiency.

Article [34] focuses on quality control in manufacturing with the main objective of improving quality control processes by integration of machine learning. The authors train a machine learning model that predicts how likely a machine is to produce faulty products. The actual predicted value is the generalized defect level (GDL), which, in simple terms, is one of 10 groups that determine how many defected parts a machine produces. Quality management is not present in Table 2-1 and in a further look, the Odoo ERP provider mentioned in the previous article [33] has a quality management module for manufacturers [8] that would be a good fit for the feature covered in this research article. So, this article will also focus on a module not in the five core modules of this thesis. The authors collected data in a small manufacturing company for a month. The dataset includes data from three different systems: IoT devices, a Manufacturing Execution System (MES), and an ERP system. The ERP was responsible for collecting information about orders as well as providing part identifiers, batch sizes, and customer requirements to the dataset. The data was split into 70% training data and 30% testing data, using a 5-fold cross-validation to train and evaluate the model. After testing multiple different models, Gradient Boosting was chosen as the final model to go forward

with. The model reached an F1-score of 0,78 and the authors hypothesized that the performance would get better after gathering more data, as the first training dataset was limited in size. The article lacks an in-practice evaluation, but the article states that implementing the approach will reduce quality inspection costs and improve accuracy of defect detection in the company.

In [35], a machine-learning based predictive analysis approach is introduced for a very specific problem. In textile industry, cloth waste must be taken into account when ordering materials for a customer order. In the company that this study was made for, the waste rate was a fixed number that was added to the number of materials ordered. Inaccurate waste rates create over or under production. This is why the authors train multiple machine learning models to predict the amount of waste a new order would have. The article states that the problem area is resource management, and this can be found in Table 2-1 under the project management module as feature F3.2. The training data for the models was acquired from the company's ERP system called Canias and it included order and production process information. The data was from January 2021 to September 2021 and included 1018 datapoints and 20 features, and was divided into two sets, 90% for training and 10% for testing. The four machine learning methods used were support vector machines, decision tree, random forest, and bootstrap aggregation which is also called bagging. The bagging model had the best results with an R-squared value of 0,86 and a mean absolute error of 1,71. Interestingly, they also evaluated feature importance and ranked the five most important features from the database. These included order amount, colour of the product, fabric mixture and attribute, and the cutting centre used. For the benefits of integrating the model into the company's workflow, the authors shortly state that the company will achieve better production results with the machine learning approach as more accurate resource management decisions can be made.

The article [36] is focused around implementing an ERP system for electric vehicle battery recycling. It goes into detail about how the ERP is designed, but most importantly for this thesis, the ERP is enhanced with a machine learning model that predicts the supply of electric vehicle batteries. This is called predicting future market supply in the article and is done so that it would be easier to respond to risks like market sensitivity. In this case determining the module and feature where this predictive analytics is used is difficult. The article does not mention where the model integrates into the ERP system and for a recycling business, predicting the battery supply is close to sales forecasting. However, as a highly exceptional

case, this is left uncategorized and therefore not mapped to Table 2-1. As for the forecasting model in question, linear regression, decision trees, and support vector machine were chosen as the machine learning methods for it. The author also lists the Python libraries used to build the models: Pandas, NumPy, Scikit-learn, Matplotlib, and Seaborn. The dataset consisted of data gathered from many official sources including the number of registered cars, population of the country (Jordan), and GDP. The number of electric batteries supplied to the recycling centre is then counted based on a formula. The dataset was split into subsets of 70% training data and 30% test data. The linear regression model greatly outperformed the other models, having the best mean squared error of 0,62. The author states that integrating the forecast of battery supplies into the ERP will help with manufacturing plans and developing infrastructure as the future supply can be considered.

The authors of [37] study how machine learning models can be used in predicting market demand in the pharmaceutical field. This study is a bit different from the earlier studies that have done market demand predictions, as it uses sentiment analysis from reviews as the main factor in predicting the demand. The article explains that the implemented predictive model is integrated into multiple modules in the ERP. The actual predictions can be integrated into inventory management and production planning modules, but it integrates into the customer relationship management module as well because of the nature of the training data, which is gotten from customer reviews. Other studies have not been as vocal about the integrations, so this is an interesting case. Sales forecast is F4.2 in Table 2-1, but in this case, sales data is not used at all. However, the model is used in inventory management model for reordering, which is F2.1, and this study is therefore categorized to that feature. The study uses a publicly available dataset for training the models. The authors explain that the data is not gathered from an ERP originally, but data of this nature would normally be found in a customer management relationship module if an ERP would be used as the source. The data mostly consists of reviews and ratings for pharmaceutical products and has over 200.000 datapoints. Sentiment analysis for these features is used together with machine learning methods, XGBoost, random forest, support vector regression, and k-nearest neighbour, to predict demand. When evaluating model performance with mean squared error, mean absolute error, and R-squared, random forest was the best and was selected for future use. The predictive model was not integrated into an ERP in practice, but the authors made several statements about the benefits of integration, nevertheless. It minimizes overproduction and shortages,

improves operational efficiency, and as the predictions are derived from sentiment analysis, marketing strategies can be adjusted based on them as well.

The study [38] leverages machine learning for two tasks in ERPs: demand forecasting and inventory management. Demand forecasting is, again, the feature F4.2 in Table 2-1. In the article, the machine learning method implemented for inventory management is responsible for replenishment of items, which is the reorder points feature F2.1. The data is gotten from an ERP system of the sponsor company from a four-year period, and the dataset is divided into 80/20 split of training and testing data. All the models are implemented in Python programming language. Multiple machine learning methods are used for demand forecasting: random forest, support vector regression, wavelet-artificial neural network, and wavelet-long short-term memory (W-LSTM). The methods are evaluated based on multiple statistical methods, but also the KPIs of the company. In academic terms, a statistical model called simple exponential smoothing performs best, but considering the KPIs, W-LSTM is mentioned to perform better, resulting in less stockouts for example. The best model for the task could not be chosen, as each has their own drawbacks which must be considered in the company. For inventory management, the machine learning methods chosen for the task were Q-learning and deep Q-network. In this case, Q-learning performed better in terms of profit, but it might be because the deep Q-network requires significantly more learning data. The authors do not discuss the benefits of integrating these models, but the inventory replenishment model is beneficial profit-wise as could be seen after evaluating the models against the company's KPIs.

In [39], processes of a manufacturing company producing sanitizing products are optimized using a digital twin framework together with machine learning methods. The authors attempt to predict several things: material consumption, delays in laboratory analysis, and reactor allocation. The data used in the study was gathered from the supervisory system of the company, the manufacturing execution system and an ERP system. The ERP was responsible for providing inventory data, product recipes, material bills, and batch reports of the produced goods. The implemented models are not integrated into an ERP system in the study, but the data gathered for them would most likely be gathered from inventory management and financial modules. However, the features discussed in the paper are not included in Table 2-1. Like before in the review, the programming language Python is used, together with its libraries Scikit-learn, Pandas, and Numpy. The implemented machine learning methods were random forest, gradient boosting regressor, decision tree, ridge classifier, logistic regression,

and k-nearest neighbours. For training these models, the data was divided into 70/30 split of training and test data. As the models were both classifiers and regressors, there were two sets of evaluation methods too. For classifiers, accuracy, f1-rank, and ROC-AUC were used and for regressors, mean absolute error, root mean squared error and R-squared were used. Random forest classifier was the best method for predicting which reactor was the most optimal for any product type and material combination. Predicting the loading time of a batch had the most success with gradient boosting. Quality issues in the processes were predicted by predicting delays in the laboratory analysis. This had good results as well, but the authors do not say which machine learning method was used in this task. A random forest regressor was trained for predicting material consumption, but this had varying results based on the material. The benefits from the predictive analytics from this study come from process optimization. Mainly, the authors stated that the models help with sustainability by minimizing waste in materials and energy as well as reducing rework in the manufacturing processes.

Authors of [40] train multiple machine learning models to predict the quality of fabric before a complex manufacturing process is undertaken. Eventually, the best model would be used in practice to adjust the loom based on the prediction. Like in [34], this study is focused on quality which in ERP systems is handled by the quality management module. As the module is not of core focus in the thesis and not present in Table 2-1, the feature enhanced with predictive analysis in this study is left uncategorized. The data gathered for the study includes information from IoT devices and an ERP system. The dataset includes manufacturing parameters, yarn types and their applications, and data about quality. Overall, the dataset has over 350000 rows and 52 features. Interestingly, for the first time in this literature review, automated machine learning (AutoML) is used. This means that the authors are free from the actual training task and even the definite machine learning method is not selected; the AutoML tool will handle it for them. They use seven AutoML tools: AutoGluon, EvalML, AutoViML, FLAML, H20AutoML, PyCaret, and TPOT. To compare the tools fairly, configurations like hyperparameter optimization technique, training duration and so on, were kept the same for all of them. The models were evaluated by four metrics: Mean absolute error (which was also selected as the one measure for selecting the best model), R-squared, mean squared error, and mean absolute percentage error. EvalML was chosen as the best model, as it reached the best score on mean absolute error. However, AutoGluon was a strong candidate as well, as it performed best in the other metrics. It could not be chosen to be used in practice though, as the time to generate a prediction with the model made with AutoGluon

was too long, halting the processes for five minutes at a time. The study concludes that the fabric quality prediction will help companies optimize processes for minimizing quality defects and as said earlier, adjusting the looms was one of these measures. This will result in less waste of materials and need for rework, which leads to cost savings. Eventually, optimizing processes this way will lead to greater quality, better products, and higher customer satisfaction.

Article [41] is a rather short study about the problem of predicting lead times in manufacturing processes with highly customizable orders. As discussed previously in this literature review for [29], this feature would also fall under a manufacturing execution system (MES) module in the ERP which is not included in Table 2-1. Interestingly, the dataset is meant to be created from both an ERP and, in this case, a standalone MES. The ERP will provide all the parameters of the order, and the MES will provide data on the current state of the factory which includes, for example, availability of machines and workers. Adding the two data sources together resulted in 32 features. However, the study has a limitation. It only trains the models on simulated data, which does not show the models' true capabilities in real life scenarios. Like in [40], AutoML is used for training the models. Instead of using multiple AutoML tools, the authors only use the H20 AutoML which was also used in [40] as one of the tools. Root mean squared error was used as the evaluation metric, and from all the different machine learning methods, gradient boosting had the best results. Deep learning methods were excluded from the pool of possible models, however, as it would have taken too long to train them. The introduction of the article explains that scheduling methods for this kind of manufacturing already exist, but they all require that the lead time is known. The implemented predictive model will help in order scheduling as estimating the lead time makes these methods more effective.

3.3 Discussion

RQ1 is focused on discovering which parts of an ERP machine learning-based predictive analysis could be integrated into. For this reason, the articles chosen for the literature review were mapped to Table 2-1, which includes modules and their features. These results can be seen in Table 3-4, which acts as the answer to RQ1 from the point of view of the literature review.

The thesis focuses on five core modules that are most relevant to the partner company, but this restriction was not enforced for the selection of articles in the literature review. Table 3-4

includes these five core modules, and other modules mentioned in the articles are grouped under “other known modules”. These modules could be named as well, but they are not of interest for this thesis and will not be discussed further. Furthermore, when it could not be determined which module the machine learning model was integrated into, it got labelled as unclassified. There were some difficulties in determining modules. Some articles did not explicitly state which module the developed model integrates into, and some articles did not even mention a module at all. The module was determined by the nature of the functionality described in the article if information was otherwise limited.

A machine learning model could be made for multiple features. In these cases, the model was mapped to both features. This means that Table 3-4 has more occurrences of features than there were reviewed research articles. An example of this was F5.8 and F5.9, which could be simply grouped together as budgeting. The most glaring issue when mapping the models to features was that a feature was put under a different module in Table 2-1 than what the article clearly says the model is integrated into. In Table 3-4, the feature was mapped under the module that was mentioned in the article. This led to F4.2 being in two modules: inventory management module and sales module. Even though sales forecasting (F4.2) could be more generally said to be a feature under the sales module by itself, it was clear that in literature the forecast was most often integrated into inventory management module’s workflow.

Table 3-4. Modules and features ML-based predictive analytics are integrated into in literature

Module	Feature	Times used
Human resource management module	-	0
Inventory management module	F2.1: Reorder points	2
	F2.2: Safety stock evaluation	1
	F4.2: Sales forecasting	4
Project management module	F3.2: Project resource management	1
Sales module	F4.2: Sales forecasting	1
Financial module	F5.8: Management of funds	1
	F5.9: Management of costs	1
Other known modules	Features outside of Table 2-1.	6
Unclassified module	Features outside of Table 2-1.	2

RQ2 is about what benefits can be gotten from integrating machine learning-based predictive analytics into an ERP. Table 3-5 gathers answers from the literature to provide the answer to

this research question. Only 13 out of the 17 reviewed articles are present in the table as not all articles mentioned any benefits at all. The table is constructed from benefits that were clearly stated in literature. It could be possible to generalize the benefits predictive analytics had in the four articles left out of the table, but that would be rather speculative. It is important to note that these benefits can be situational, and the models are often customized for a single company. The results and, therefore, benefits can vary.

The benefits generated for the company by the integration of ML-based predictive analytics into ERP systems is not the focus in literature. The reviewed articles were most interested in proving that the machine learning model reaches a high accuracy. A reason for this could be that the benefits are already gotten through the processes that the models are made to optimize, not necessarily from the models themselves. It still does not mean that the benefits could not be studied more; it would have been more interesting to know how much the processes were optimized. Most articles only theorized what benefits the models could provide and only a handful tested the models in practice to provide quantitative evidence of the benefits. This is understandable, as it requires a partner company to test the models in practice. The effect of some benefits is also more difficult to study than others. For example, it is simpler to count the number of products sold than to determine how much customer satisfaction has improved.

Numerical data about benefits was provided by four articles in total, all of which did sales forecasting. Overall sales were improved by increasing products sold from an average of 166 to 211 per month [23]. Inventory management was improved by reducing inventory stockouts by 25%, overstocking incidents by 18%, and holding costs by 15% [28]. Furthermore, inventory turnover was improved by 25% [32] and 230% [33].

Table 3-5. Benefits of machine learning-based predictive analytics in literature

Benefit	Articles
Reducing risk	[23]
Increasing products sold	[23]
Increasing customer satisfaction	[23], [31], [33], [40]
Reducing inventory backlog	[23]
Reducing inventory safety stock	[23]
Optimizing inventory stock levels	[27], [28]
Reducing inventory stockouts	[28], [37]
Reducing inventory overstocking incidents	[28], [37]
Reducing inventory holding costs	[28], [33]
Reducing downtime costs of machinery	[29]
Increasing inventory turnover	[32], [33]
Reducing inventory operational costs	[33]
Increasing operational efficiency	[33], [37]
Reducing quality inspection costs	[34]
Improving defect detection	[34]
Improving production results	[35]
Reducing rework in manufacturing processes	[39]
Minimizing waste	[39], [40]
Minimizing quality defects	[40]
Improving order scheduling	[41]

There were many machine learning methods used in the literature, and they were pointed out in the review to give perspective for the approach in the case study of the thesis. Overall, 20 different methods could be identified from the reviewed literature. All the identified methods are listed in Table 3-6, together with how many times they were mentioned as an option and how many times they were ultimately used in the final model. Not all articles had a single model as the end product, so for these articles the model chosen as the best in the article was counted as the used one.

All variations of a model were counted as a single group in Table 3-6. For example, article [3] used a support vector machine (SVM) together with particle swarm optimization (PSO), and this was counted towards SVM. Also, SVM was referred to as support vector regression in [37] and [38] contrary to the other articles, even though all the SVM models trained in the literature were used as regressors, so they were counted towards SVM as well. Many articles also used classical techniques in addition to machine learning. The most prevalent of these was Autoregressive Integrated Moving Average (ARIMA) which is a statistical forecast model and not considered machine learning [38]. These classical techniques were used as a

basis for comparing the performance of machine learning methods. They were left out of Table 3-6, which only includes ML methods.

Some articles only chose one model to be tested based on past literature or personal preference, whereas other articles compared multiple methods between themselves and then chose the best one to use. For example, when compared to other models, SVM was only selected once to be the best fit. It did not seem to perform that well in predictive analysis in ERPs compared to other methods. In contrast, Random Forest performed excellently in multiple articles and was chosen as the best fit a total of five times, even though it was also compared to other models every time. LSTM was one of the models that were not always compared to other methods. Two out of three times it was chosen as the best fit by the authors before even training the model.

Table 3-6. Machine learning methods used in literature

Machine learning method	Times tested	Times used	Times compared
Random Forest	7	5	7
Support Vector Machine	5	1	5
Decision Tree	4	0	4
Long Short-Term Memory Neural Network	3	2	1
Gradient Boosting	2	2	2
AutoML	2	2	0
Linear Regression	2	1	2
XGBoost	2	1	1
K-nearest Neighbours	2	0	2
Multilayer Perceptron	1	1	0
Backpropagating Neural Network	1	1	0
Bagging	1	1	1
Q-Learning	1	1	1
Wavelet Long Short-Term	1	1	1
Wavelet-artificial Neural Network	1	0	1
Naïve Bayes	1	0	1
Deep Q-Network	1	0	1
Ridge Classifier	1	0	1
Logistic Regression	1	0	1
Ensemble of Random Forest and XGBoost	1	0	1

The most used machine learning methods in the literature review were Random Forest, Support Vector Machine, Long Short-Term Memory, and Decision Tree. These were used

most frequently. Only two of them, Random Forest and Long Short-Term Memory, were also used most of the time. The other two were not accurate enough to be chosen as the best fit in the articles using them. It is also interesting to note that there are 11 methods out of the 20 that were only used a single time. A conclusion can be drawn from that; these methods are not as generalizable as the most used ones and are more or less situational.

4 Machine Learning-based Predictive Analysis for Budgeting District Heating Network Projects

In this chapter, a case study is conducted to integrate machine learning-based predictive analysis to an in-house ERP system which has been built during the writing process of this thesis.

4.1 Background

The partner company of this thesis is a small business that builds district heating networks in the Helsinki metropolitan area. The company has started developing its own ERP system and has recently migrated from its old ERP provider to the new in-house ERP. This brings a lot of new possibilities for the company. Previously, the old system was not customizable and did not fully meet the company's needs but with the new system in place, new features could be added that provide value for the company.

Quickly, the idea of adding machine learning to the system was brought up. Training machine learning models for the company's use is now a lot easier, as all data gathered by the ERP is readily available in a single database. And as mentioned in Chapter 1, enhancing an ERP with machine learning-based predictive analysis improves the competitiveness of a business [6]. In fact, improving competitiveness is exactly the reason behind implementing the in-house ERP system in the first place.

The business model of the partner company is largely project oriented. The company has multiple worksites and projects around the Helsinki metropolitan area for multiple customers simultaneously. In the literature review, a project-oriented view for machine learning-based predictive analysis is completely missing. Most of the designed machine learning models are built for manufacturing businesses. Only one article had a model integrated into a feature of project management module, that feature being resource management, and even that one was for a manufacturing company. Even though a model integrated into the inventory management module was popular in the literature review and, for example, a model predicting reorder points for parts and materials needed in district heating network projects would be beneficial for the partner company as well, a project oriented perspective would not only be more appropriate for the partner company but would also address the gap in the literature.

Candidates for a machine learning solution for improving project management were discussed in multiple meetings of the ERP system team. Two promising options were determined:

scheduling and budgeting new projects. Project schedules and budgets must be considered every day in each project, making them very important to get right. A model predicting the budget of a project would be important during the project to plan costs, and it would work as a reference when making offers to customers. However, implementing the model would be difficult. The accuracy of the model would depend on current pricing lists of the company, as if these are not taken into consideration, the model would become inaccurate after the pricing of materials, parts, or labour changes. The problem is that these pricing lists are not readily available and would have to be created and modified to a form that the model can be trained on. For this reason, a model predicting the schedule of a project would be simpler to develop. Precise scheduling does not provide the company with as much value as budgeting, however. Most of the projects are made for familiar customers and are short in length, which means that their timeline is easier to predict by supervisors too. The only issue for scheduling lies in large projects that are rarely conducted, which can have contractual penalties for crossing deadlines.

Because of the time frame set for this case study, it would only be possible to make one of these models. Finally, it was decided that the model built in this case study is trained to predict the budget of projects. Even though it is more complex, budgeting is the most valuable one for the company from the two options presented. The literature review had no influence on the decision between the two models. A model for budgeting was trained in article [3] but the budget of the whole company was predicted in that study and is, therefore, not applicable to this case study. Scheduling had a similar situation, article [41] having predicted lead-times of manufacturing processes. Again, the study is not directly applicable to the project-oriented approach studied in this case study.

4.2 Data

The training data for the model is gathered from multiple sources. The newly developed ERP system is of no use at this moment, as it has been in use for so little time that meaningful amount of data has not been accumulated to it yet. Instead, the data is gathered from both the old system and the personal computers of project managers. The model will use projects from 2023 to 2024 as the training data.

There are four pieces of data included in the training data of the model. The measurement reports of project worksites are gone through in Section 4.2.1, project budgets in Section 4.2.2, basic project information in Section 4.2.3, and publicly available cost indexes in

Section 4.2.4. In addition, feature engineering and data augmentation on the whole dataset is gone through in Sections 4.2.5 and 4.2.6 respectively.

4.2.1 Measurement Reports

An important part of the training data is measurement reports. A measurement report is a detailed report of all the work done for a project. In the context of district heating, the report includes work like welding and cutting pipe, length of pipe installed, and the number of branches in the network. In practice, the reports are Excel sheets that are filled in by project managers after a project is finished. In the reports, all tasks and materials use standardized names and units of measurement, which makes the data easily usable for the model. The shape of the data in the Excel worksheets can be seen in Figure 4.1.

	Pipe Size 1	Pipe Size 2	...	Pipe Size N
Work Type 1				
Work Type 2				
Work Type 3				
...				
Work Type N				

Figure 4-1. The format of measurement reports

Measurement reports are excellent training data for the model. Project budget is heavily dependent on the amount of work and materials needed for the project. The reports have both. However, a measurement report is only written after a project is finished, but the budget prediction should be done before a project has been started. Including everything from the measurement reports would introduce data leakage, where the model is trained on data that is not available at inference-time [43]. For this reason, only the work types that can be easily estimated when a project is being planned are included in the training data from the reports. For example, the number of welds across different sizes of pipes can be difficult to estimate beforehand. But estimating the length of pipe needed for the area or the number of branches the pipeline will have is a lot simpler as the customer will hand in initial plans for the district heating network which can be used to count these work types.

There were measurement reports from as early as 2020, but because of the large number of reports to process and the limited timeframe of the study, only reports for projects from 2023 to 2024 were used in the model. In this time range, there are 473 measurement reports.

Measurement reports had about 35 different types of work. Of these, 10 were determined to be easily estimable. Extracting these work types from the reports was made simple as there were only two templates used for reports in the company. This made it possible to simply see which row holds information for the work type to be extracted. Determining the column was more difficult: each work type is marked to the report based on the size of the pipe that is installed or worked on. This means that each work type also has as many variants as there are sizes of pipe. The data for each work type included in the model was therefore needed to be dynamically extracted by joining the work type to the size of the pipe written on the first cell in the column.

After processing the reports, the data had to be joined to the training data. This was the most difficult part of the process. Before the new system, there were no project numbers that could be used to identify projects in a simple way. Instead, the measurement reports as well as projects were named after the address of the worksite. Sometimes the address written to the report would be directly found as a project as well, but other times the address would be written slightly differently, or a different address was used altogether. In these cases, Google Maps was used to search for projects in streets close to the address. Some addresses also had had multiple projects throughout the years. To choose between multiple projects, the date of the report file's latest save was used to determine the project, as that could be compared to the projects' schedules.

Of the original 473 measurement reports, 381 were usable in the training data. 42 were unusable as it could not be determined which project the report was made for, and 50 were unusable because the reports were made for smaller worksites that did not have a project registered to the old system at all.

4.2.2 Budget

The model will predict the budget of a project. In this case, the budget is defined to be all the costs that can be targeted to the project. As these total costs are not gathered anywhere nor divided by project, they are calculated by summing up the invoices sent to customers during and after each project. However, this would not result in the actual total costs, because the invoices also include the profit margin. The profit is subtracted from the total based on a fixed margin chosen together with supervisors.

The invoices were received from the CEO of the company directly. There were 548 invoices from 2023 to 2024. Around 50 of the invoices were in a different format in which an invoice included costs for multiple projects. The other invoices were only for a single project each. This is also why the invoices were processed in two batches. Most invoices were converted from PDF-files to text files using the *pdfplumber* library in Python. As the format in invoices for the first batch was the same for all, the end sum of an invoice was possible to be extracted using the wording around it after converting the invoices into text files. The second batch of invoices had their end sum extracted manually for each project individually. After subtracting the profit margin, these end sums were used as the target variable in the machine learning model.

All project costs were mapped into the training data by the address used in the invoices, similar to how measurement reports were handled. The addresses were used as the worksite on the invoices and were easily found in the old system as projects, so there was no difficulty getting the project costs into the training data.

4.2.3 Project Information

The old system is used to export all basic project information that is to be used in the training data. These include the start date, end date, project manager, and customer of the project, as well as the address of the worksite. All this information can influence the budget of a project. The start and end dates are straightforward as longer projects usually have a bigger budget and building district heating networks in winter months have extra costs. The project manager is relevant because some managers can take up more challenging projects than others, or some teams can be more efficient. The customer of the project also matters because projects might have unique challenges based on the processes and requirements of the buyer. Lastly, the location of the project can drive up costs as well. Building in the city centre is more expensive and worksites far from the company's warehouse cause more costs in logistics. These are only some reasons to include this data, but the model can find even more connections between the budget and project information. A summary of the attributes in basic project information can be found in Table 4.1.

Table 4-1. Summary of basic project data

Attribute	Description
Start and end dates	Project's schedule. Only year and month are included. Months are encoded cyclically. Duration in days derived from the dates as a new feature.
Project manager	Manages the project. Name is converted to the user's ID in the new ERP system, and the feature is one-hot encoded.
Customer	The buyer.
Address	The location of the worksite. Only the postal code is included in the dataset and one-hot encoded.

In 2023 to 2024, there were 454 projects registered into the old system. All these projects were kept in the dataset at first but were later dropped if measurement records could not be found, or the budget could not be calculated for them. In addition to filling in missing values like end dates and addresses of projects, the data processing included some interesting steps. Before that, names of project managers were mapped into their respective user IDs in the new ERP system for easier integration later. Start and end dates were converted into an additional feature of project duration as well as into a year and a month as separate columns. The months were then further encoded cyclically with sin and cosine functions, which improves the model's ability to understand the cyclical nature of months [44]. Only the postal code was extracted from the addresses, as locations more detailed than that were deemed too precise to affect the budgets of projects for this company. At first, the city was also extracted from the address, but the final dataset had projects only from a single city making the column irrelevant. Furthermore, all categorical data was one-hot encoded.

4.2.4 Cost Indexes

To avoid the time-consuming task of creating pricing lists for all labour and materials used in the projects, publicly available information about costs will be integrated into the model to keep it from declining in accuracy as time passes. This includes material and labour costs. Without taking these costs into account, the accuracy of the model will slowly decline as time passes and costs increase.

The two indexes used are labour cost index and cost index of civil engineering works. Only the indexes for the company's business sector are used. Both indexes are available publicly in Statistics Finland [45]. The first will account for payroll, while the second will account for

costs more generally. These indexes were gathered for the time range encompassing the training data, and when the model is used in the ERP system, the index can be gotten through the application programming interface (API).

However, the current indexes are not available because of the publishing schedule. For labour cost index, the data is published quarterly, and the index of each quarter gets published only two quarters later. The cost index of civil engineering works gets published monthly and is off by two months. This means that the training data will have to use old indexes too, because when the estimation is done with the model, the current index will not be usable. As the labour cost index is highly seasonal, having the highest values during summer and lowest values during winter, the training data will use the index for the same quarter from the previous year. That is arguably the correct option, as the change in index values was more volatile between subsequent quarters than quarters a year apart because of the seasonal variance. For the cost index of civil engineering works, the same seasonal variance was not observed. For this index, the training data will use the index's value from two months before the project start date.

4.2.5 Feature Engineering

Start dates and end dates of projects were already converted into a single new feature, duration. This belongs in the base dataset as the exact start and end dates are not that usable for the ML models. The duration of a project is an important part of the training data. Other features in the dataset are usable as they are, but mainly in the measurement report data the connections between the many pairs of pipe size and work type can be difficult for a model to derive. For this reason, additional features were created from measurement report data to help in the learning process. Later, each of these features were tested in isolation if they helped the models or not. All the beneficial features were then added as a single new dataset to test with, in addition to the base dataset which only included the original features. The new features created from measurement report data are summarized in Table 4-2.

There were five different feature sets created from the measurement report data. Firstly, the amount of work in each separate type of work. As discussed earlier, the measurement reports included multiple work types across all different pipe sizes. This results in scattered data where a single pair of work type and pipe size might only have a few values in the whole dataset. This new feature set was made by summing up the values in all columns made for each work type.

The second feature set is a small deviation from the previous one, as summing up each type of work regardless of the pipe size misses a key detail. District heating networks with larger pipes also cost more to make. The second feature set takes this into account by summing the work but weighing the amount of work by the pipe size. This way, larger pipe sizes amount to more work done, which reflects how they affect the budget.

As a single new feature, all measurement report data was summed together to form the total amount of work done in a project. This feature is straightforward. Each column representing a pair of work type and pipe size were added together to form a single number.

The final two additional features were created by thinking about the measurement report data as a whole. For example, many different work types could lead to a difficult or more complex project, which could in turn increase costs. This resulted in a new feature for the count of distinct work types. Similarly, pipe sizes could drive up costs. This is already known as larger pipe is more costly, but a project using mostly smaller pipe with a tiny section of larger pipe might still require setting up for larger machinery. This resulted in a new feature of the maximum size of pipe in a project.

Table 4-2. Feature engineering summary

Feature	Description
Amount of work for each work type	Instead of including measurement report data as pairs of work type and pipe size, these new features use the type of work as the sole differentiating factor.
Weighted amount of work for each work type	Similar to the previous feature set but weights the amount of work based on the pipe size.
Total amount of work	Total amount of work across all work types and pipe sizes.
Count of distinct work types	The amount of different work types used in a project.
Maximum pipe size	The largest pipe size used in a project.

4.2.6 Data Augmentation

Data augmentation was already discussed in the development team of the company when it was decided that a machine learning model would be trained to predict project budgets. The company was leaning towards using neural networks as the machine learning approach, which benefits from a large dataset. However, the reality was that there were not many projects

conducted in the company, and the dataset was going to be quite small. Augmenting data was needed especially because of the small number of data points.

While data augmentation is already widely used in image classification, it has not been studied as much with tabular data as it's often more complex and the data can have different structures based on the context [46]. This means that there is also no reliable tool that can be used to automatically augment realistic training data. For this reason, data augmentation was done manually using domain knowledge.

Augmentation must be done deliberately so that the augmented data would resemble realistic projects. For example, adding noise to the features of existing projects was ruled out as a possible approach because this could lead to situations where the noise lessens the amount of work done in the project but simultaneously increases the budget of the project, which is not realistic. Instead, it would be important to think about which operations on the existing project data would lead to new projects that could have been done in the company but still being different enough from the existing training data for the models to benefit from the larger dataset. With this in mind, two styles of augmenting data were implemented.

The first data augmentation method was to simply scale the projects. Slightly scaling all the work done in a project and then scaling the duration and budget of the project similarly could possibly lead to realistic, augmented projects. When a project includes more work, it also costs more and takes slightly longer to carry out.

The second data augmentation method was to create new projects by pairing different parts of project data between them. By taking all measurement report data from one project, and basic project data from another project, a new realistic project could emerge. Considering that an extremely short project could not possibly include all the work of a large project, this process was not done randomly. K-Nearest Neighbours was used to ensure new projects were created only from two projects that were already similar to each other.

It is important to note that validating a machine learning model with augmented data can skew the results. Instead of crafting augmented datasets from the base dataset, only the training data was augmented during cross validation.

4.3 Training the Model

Two machine learning models were trained and evaluated in the same manner for this case study. Multi-Layer Perceptron (MLP) was chosen as one of the models to try purely based on the company's preference and Random Forest was chosen as the alternative model to MLP as it was the most used model in the literature review.

Both models were trained and evaluated with K-fold cross validation. The final dataset only had about 200 data points, so the number of folds in cross validation was left at five to utilize most of the data for training the models with the small base dataset and simultaneously keeping the training time manageable for the larger augmented datasets.

After training, the models were evaluated and compared with two evaluation methods: R-squared and mean average error (MAE). From these two, MAE was only used internally to quantify the model's accuracy as it gives a monetary value. As such, it is not relevant without disclosing the actual project budgets in the training data and is therefore not used in the case study. Because of that, R-squared will be used as the sole evaluation metric in this thesis.

The models were trained with the Python library *scikit-learn,* and plots were made with libraries *seaborn* and *matplotlib*. Both models had their best configurations tested on the base dataset, base dataset with chosen features from feature engineering, and with both styles of data augmentation.

4.3.1 Random Forest

The Random Forest regressor from *scikit-learn* was used to train a Random Forest model for predicting project budgets. Four hyperparameters were tuned for the model: the number of trees (*n_estimators*), maximum depth of the tree (*max_depth*), minimum number of samples required for leaf nodes (*min_samples_leaf*), and the maximum number of features to use (*max_features*). The other parameters were left at their default values. All tested hyperparameter values are listed in Table 4-4.

Table 4-3. Tested hyperparameter values of the Random Forest model

Hyperparamater (in scikit-learn)	Tested values
n_estimators	10, 50, 100, 200, 300, 500, 1000
max_depth	2, 3, 5, 7, 10, 15, 20, none (no restriction)
min_samples_leaf	1, 2, 3, 5, 10, 15, 20
max_features	'sqrt', 0.3, 0.5, 0.7, 1.0

The Random Forest model was also used to determine which of the new features from feature engineering had a positive impact on the training process. Each of the features were tested separately, and if they resulted in a better R-squared than the base dataset, they were later included in the dataset to be tested together. The final dataset with all the features included was tested on both models.

4.3.2 Multi-Layer Perceptron

The Multi-Layer Perceptron model was trained using a Multi-Layer Perceptron regressor from *scikit-learn*. Multiple different architectures for hidden layers were tested, and those can be seen in Table 4-3. Other than that, the configuration of the regressor used default values, except for using 2000 as the maximum number of iterations to decrease training time for easier testing of configurations.

The hyperparameters of the MLP model were not tuned as much as for Random Forest, as the size of the dataset quickly emerged as the sole problem for its poor performance. Instead, efforts were concentrated towards data augmentation methods that were mostly aimed at increasing the accuracy of the MLP model.

Table 4-4. Tested hidden layer configurations for the MLP model

Number of hidden layers	Tested configurations (number of neurons in hidden layers)
1	(4), (16), (64)
2	(64, 32), (128, 64)
3	(128, 64, 32)
4	(256, 128, 64, 32)
5	(256, 128, 64, 32, 16)

4.4 Results

In this section, the Multi-Layer Perceptron and Random Forest models are trained and evaluated as explained in Section 4.3. The models go through similar steps, but the results are gone through individually. The results are discussed in depth in Chapter 5 together with other results of the thesis.

4.4.1 Random Forest

The Random Forest model was trained using K-fold cross validation with five folds. However, it was quickly noted that there was some inconsistency with this approach, as the variance in R-squared between folds was high. Shuffling the rows before dividing them into folds helped a bit, but the same variance could still be seen. The hypothesis was that the few large projects in the dataset were causing it. The most reliable result was gotten after repeating the K-fold cross validation with different random states, and therefore, 20 repetitions were made with 5 folds every time the model was trained. The Random Forest model was quick to train with such a small dataset, so repeating it was not too time consuming. The same could not have been done with the MLP model, as the training for that took significantly more time.

Each of the four hyperparameters were tested individually and the results can be found in Figure 4-4. These results gave a smaller set of combinations to test for. Many combinations had close to the same R-squared value, two best models having the same R-squared value of 0,412. The one with smaller mean average error was chosen as the best model, which was *n_estimators*=200, *max_depth*=7, *min_samples_leaf*=3, and *max_features*=0,3. These tests were done on the base dataset without data augmentations or feature engineering. The best five models and their results can be found in Table 4-5.

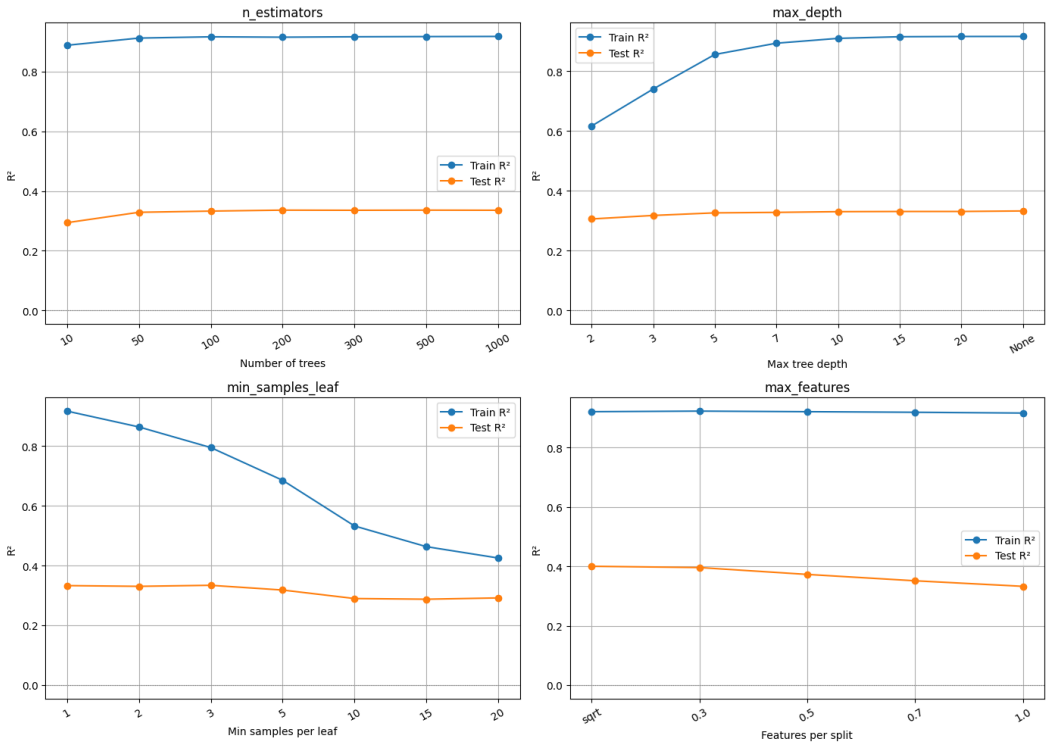

Figure 4-2. Results of hyperparameter tuning the Random Forest model

Table 4-5. Five best combinations of hyperparameters in the Random Forest model

Hyperparameter configuration	R-squared
n_estimators=200, *max_depth*=7, *min_samples_leaf*=3, *max_features*=0,3	0,412
n_estimators=300, *min_samples_leaf*=3, *max_features*=0,3	0,412
n_estimators=200, *min_samples_leaf*=3, *max_features*=0,3	0,411
n_estimators=200, *min_samples_leaf*=2, *max_features*=0,3	0,409
n_estimators=200, *min_samples_leaf*=2, *max_features*='sqrt'	0,404

After determining the best hyperparameters for the model, features hypothesized in Section 4.2.5 were tested by adding them to the dataset separately. The impact to R-squared for each of the features can be seen in Table 4-6. Three of the five features improved the model, and these were then added to the dataset for training the final model.

Table 4-6. R-squared after adding each feature from feature engineering separately

Tested feature	R-squared	Difference to base dataset
Base dataset	0,412	
Amount of work for each work type	0,475	+0,063
Weighted amount of work for each work type	0,452	+0,04
Total amount of work	0,445	+0,033
Count of distinct work types	0,405	-0,007
Maximum pipe size	0,406	-0,006

The attempted data augmentation techniques explained in Section 4.3.6 did not improve the model. The more data there was generated using these techniques, the smaller the test R-squared got. Similarly, the train R-squared got higher, meaning that the model started overfitting to the training data when more data was added. Neither of these techniques could be used in the final model for Random Forest. The results of data augmentation can be seen in Figure 4-5.

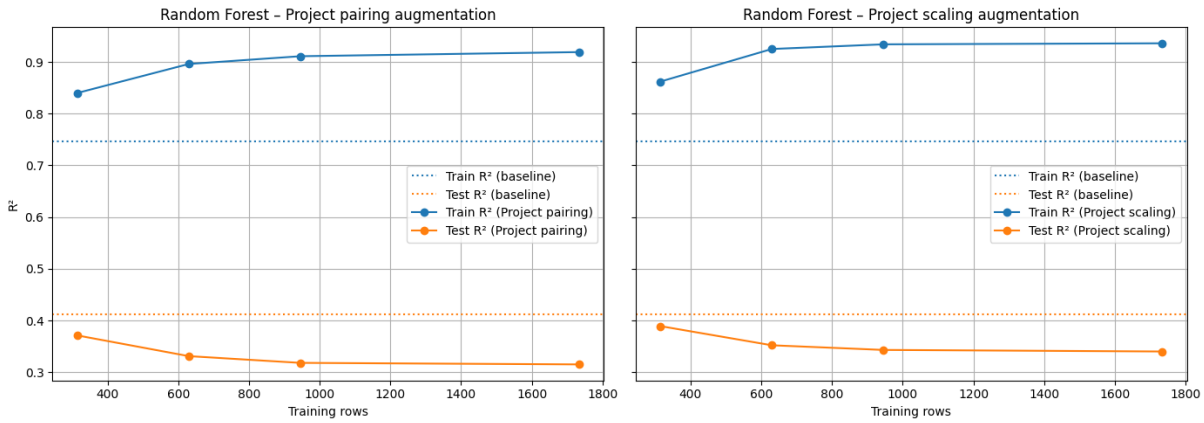

Figure 4-3. The results of using data augmentation techniques with Random Forest

The best combination of hyperparameters for the Random Forest model was the number of trees as 200, minimum number of samples required for leaf nodes as 3, the maximum number of features to use as 0,3 and the maximum depth of the tree as 7. With this configuration and all the features that had a positive effect on R-squared in feature engineering, the Random Forest model achieved an R-squared of 0,481.

4.4.2 Multi-Layer Perceptron

The MLP model was trained using K-fold cross validation with five folds. When testing the different configurations, it was clear that none of them were usable models. The test R-squared was negative in the simpler models already but got increasingly lower as the architecture got more complex. The best R-squared ended up being -0,715 for the model with a single hidden layer of 64 neurons. Negative R-squared means that the model makes predictions worse than just predicting the average every time and it cannot be used to determine how bad a model is [47]. Therefore, no further conclusions can be drawn from the increasingly lower R-squared. The R-squared of the training set, however, stayed negative at first but jumped to over 0,9 when two hidden layers were used and perfectly overfit to 1 when using three or more hidden layers. The testing results of the different architectures can be seen in Figure 4-2.

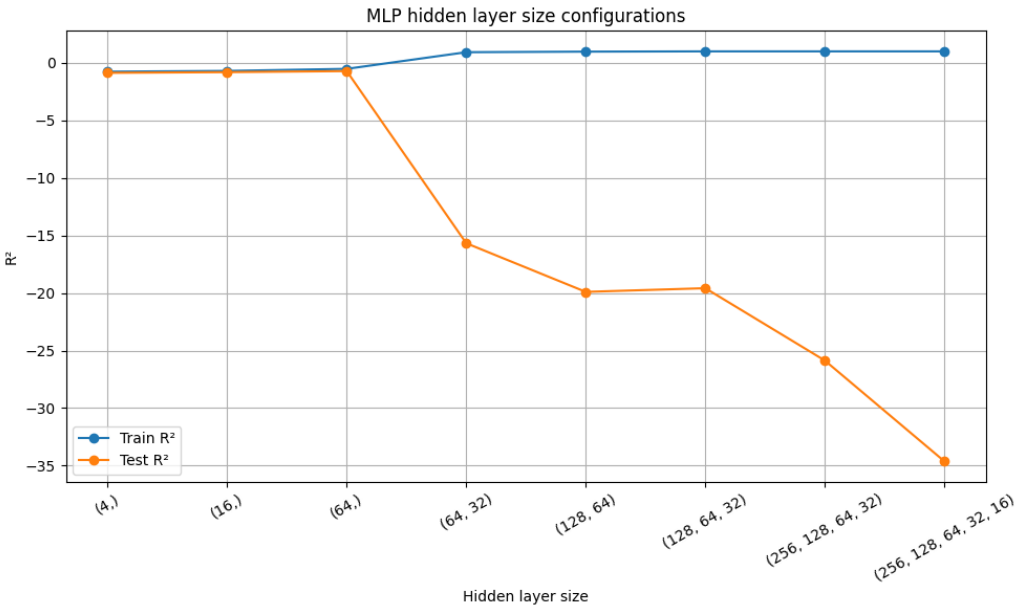

Figure 4-4. Results of testing hidden layer size configurations for the Multi-Layer Perceptron model

The data augmentation techniques explained in Section 4.2.6 were tested with the single hidden layer of 64 neurons as that had the highest R-squared of the configurations. The results from those tests can be seen in Figure 4-3. Neither of the techniques worked, and the R-squared was impacted similarly to adding complexity in the architecture with test R-squared getting increasingly lower the more data there were, and the training R-squared approaching a perfect overfit.

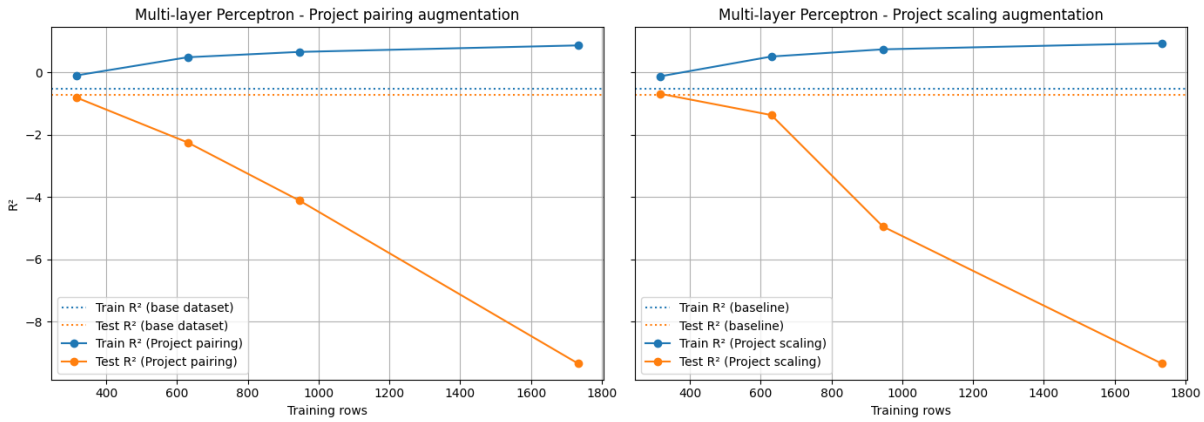

Figure 4-5. Data Augmentation on the Multi-Layer Perceptron model

After adding the features to the dataset that had a positive effect on the Random Forest model seen in Table 4-6, and testing the best MLP model with that data, an R-squared of -0,814 was gotten. This was slightly lower than the R-squared without these features, meaning that feature engineering had no effect on the MLP model.

5 Discussion

This chapter discusses results and their limitations from both the literature review and the case study, and specifically how they apply to the partner company.

5.1 Implications for the Partner Company

The models trained in the case study were not at a level that would be usable for the company without further development. The biggest problem was the data, which was difficult and time-consuming to process and limited in size. The case study showed that the data is not in a state in the company where machine learning models for predictive analytics can be created with reasonable effort. The best course of action in the future is to focus on generating easily accessible data. That is in progress, as the new ERP system is in use already, which acts as a centralized database that can be utilized in training new models after the system has produced a meaningful amount of data.

In the literature review, many studies did not use the model in an ERP system. The same is true for the case study of this thesis. The ML model for budgeting should be refined further for an integration to the ERP to be worthwhile, but integrating the model was also not possible in the timeframe set for the thesis. The absence of integration is noticeable in the case study, as the data was gotten outside of the new system, either from the old ERP or the company's management directly. This made data processing more difficult than working in a single database. With an integrated working model, it would be possible to ensure that data is easy to be input for the model to make predictions, and that the cost indexes are automatically fetched from the public APIs. The model could be integrated within the process of creating a new project or as a standalone feature into the project module.

The literature review showed that predictive analytics is most used in the inventory management module. Benefits of predictive analytics in ERP systems were also mostly found there. There is a clear difference between inventory management and project management modules regarding data. Inventory moves faster and generates more data in the process than projects. This results in easier utilization of machine learning-based predictive analytics in that module, which can be seen as more articles being written about inventory management in research literature. Conversely, project management module creates less data, and is therefore less likely to have predictive analytics integrated into the module. The company's new

attempts in machine learning-based predictive analytics in the new ERP system should be made to the inventory management module for this reason.

Even though the case study was not a success, it showed that project budgeting is a feature that can be enhanced with machine learning based predictive analytics, as the Random Forest model still managed to learn patterns from the data with an R-squared of 0,481. With further development of the model and a larger dataset size, it could reach a level where its adoption to the ERP system is possible. Furthermore, the case study showed that Random Forest was a promising model for predictive analytics in ERP systems. If new models are created in the company at a later stage, Random Forest should be experimented with again.

5.2 Limitations

Machine learning-based predictive analytics in ERP systems is very dependent on data by its nature. The outcome is often determined by what kind of data is available, and how much of it is there. This means that the results of this study, both the literature review and the case study, are not generalizable. It is possible that another study predicting project budgets in ERP systems would end up in a different outcome than the case study in this thesis, solely because of the different environment. The same ambiguousness can be said about the literature review. The benefits of ML-based predictive analytics in ERPs gathered in the literature review are a list of potential that each company can address but they must also think about if they need those benefits and if they have the resources to achieve them.

The biggest limitation of the case study was the amount of data. Ideally, this study would have been conducted a few years from now, with all the data available in the new ERP system itself instead of having to work with multiple different data sources simultaneously. That would emphasize the impact an ERP can have by bringing operations together under the same system. As of now, the case study works as a proof-of-concept as it can be said that all this data could be inside an ERP, as it is planned to integrate all these data sources there.

The literature review is somewhat limited by the selection criteria of articles. Many articles were left out because they did not propose an actual machine learning model. This naturally ruled out all articles that discussed ERPs as a whole, as creating machine learning models encompassing the whole system would be an enormous amount of work. The articles ended up mostly focusing on a single module of an ERP. It could also be possible that other similar articles would not mention an ERP at all, as creating ML-based predictive analytics in, for

example project management, is not reliant on an ERP system. This would leave the selection of articles to be studies that focused on a small part of an ERP system and had mentioned that the data was gotten from an ERP or that the model is integrated into an ERP.

6 Conclusions and Future Research

This study was conducted to understand where predictive analytics could be used in an ERP system and what benefits it could have, focusing on five core modules: human resource management, inventory management, project management, sales, and financial modules. A literature review was written for a general view of the topic whereas the case study was conducted to give measurable results to the partner company itself. These two approaches complemented each other in creating a picture of how predictive analytics could be used in the company.

For RQ1, the literature review answers it in detail in Table 3-4, showing that ML-based predictive analytics is mostly used in the inventory management module, and has some applications in other modules. The case study showed that predicting project budgets in a project management module of an ERP is a promising feature to integrate ML-based predictive analytics into, but it failed to successfully show its potential as the model was deemed too inaccurate to be used in practice.

RQ2 was answered in detail in Table 3-5 from the findings of the literature review. It lists ways in which applications of ML-based predictive analytics have optimized processes in ERP systems, again, mostly concentrating on the inventory module. The case study could not answer RQ2 as an integration into the ERP system could not be made.

Future research should expand on the case study, focusing on using more of the available data that was now left out because of time constraints on the time-consuming data processing, and attempting other ways of optimizing the machine learning models. Any further applications for ML-based predictive analytics should be made into the inventory management module.

References

[1] Z. N. Jawad and V. Balázs, “Machine learning-driven optimization of enterprise resource planning (ERP) systems: a comprehensive review,” *Beni. Suef. Univ. J. Basic Appl. Sci.*, vol. 13, no. 1, pp. 4–13, 2024, doi: 10.1186/s43088-023-00460-y.

[2] S. Katuu, “Enterprise Resource Planning: Past, Present, and Future,” *New Review of Information Networking*, vol. 25, no. 1, pp. 37–46, Jan. 2020, doi: 10.1080/13614576.2020.1742770.

[3] S. Sharma, P. Sarkar, B. Rajalakshmi, S. Lakhanpal, I. Sumalatha, and A. Joshi, “Machine Learning-based Predictive Analytics for Financial Planning and Budgeting in ERP Systems,” *Proceedings of International Conference on Communication, Computer Sciences and Engineering, IC3SE 2024*, pp. 1558–1563, 2024, doi: 10.1109/IC3SE62002.2024.10593246.

[4] N. V. Syreyshchikova, D. Y. Pimenov, T. Mikolajczyk, and L. Moldovan, “Automation of production activities of an industrial enterprise based on the ERP system,” in *Procedia Manufacturing*, 2020. doi: 10.1016/j.promfg.2020.03.075.

[5] S. Marsland, *Machine learning : an algorithmic perspective*, Second edition. in Chapman & Hall/CRC machine learning & pattern recognition series. Boca Raton, FL: Chapman and Hall/CRC, an imprint of Taylor and Francis, 2014.

[6] M. S. P. Babu and S. H. Sastry, “Big data and predictive analytics in ERP systems for automating decision making process,” *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, pp. 259–262, Oct. 2014, doi: 10.1109/ICSESS.2014.6933558.

[7] Panorama Consulting Group, “The 2025 ERP Report,” 2025. Accessed: Apr. 24, 2025. [Online]. Available: https://www.panorama-consulting.com/resource-center/erp-report/

[8] “Odoo - All Apps.” Accessed: May 30, 2025. [Online]. Available: https://www.odoo.com/fi_FI/page/all-apps

[9] Ian McCue, “ERP Modules: Types, Features & Functions | NetSuite.” Accessed: May 02, 2025. [Online]. Available: https://www.netsuite.com/portal/resource/articles/erp/erp-modules.shtml

[10] Russ Davidson, “ERP Modules Explained.” Accessed: May 02, 2025. [Online]. Available: https://softwareconnect.com/learn/erp-modules/

[11] “Core HR and payroll software.” Accessed: Jun. 09, 2025. [Online]. Available: https://www.sap.com/products/hcm/core-hr-payroll.html

[12] “Oracle Human Capital Management (HCM).” Accessed: Jun. 09, 2025. [Online]. Available: https://www.oracle.com/human-capital-management/

[13] W. Muchaendepi, C. Mbohwa, T. Hamandishe, and J. Kanyepe, “Inventory Management and Performance of SMEs in the Manufacturing Sector of Harare,” *Procedia Manuf.*, vol. 33, pp. 454–461, Jan. 2019, doi: 10.1016/J.PROMFG.2019.04.056.

[14] E. Nazemi, M. J. Tarokh, and G. R. Djavanshir, “ERP: A literature survey,” *International Journal of Advanced Manufacturing Technology*, vol. 61, no. 9–12, pp. 999–1018, Aug. 2012, doi: 10.1007/S00170-011-3756-X/METRICS.

[15] O. C. Wei, R. Idrus, and N. L. Abdullah, “Extended ERP for inventory management: The case of a multi-national manufacturing company,” *International Conference on Research and Innovation in Information Systems, ICRIIS*, Aug. 2017, doi: 10.1109/ICRIIS.2017.8002489.

[16] A. M. Atieh *et al.*, “Performance Improvement of Inventory Management System Processes by an Automated Warehouse Management System,” *Procedia CIRP*, vol. 41, pp. 568–572, Jan. 2016, doi: 10.1016/J.PROCIR.2015.12.122.

[17] M. A. Razi and J. M. Tarn, “An applied model for improving inventory management in ERP systems,” *Logistics Information Management*, vol. 16, no. 2, pp. 114–124, Apr. 2003, doi: 10.1108/09576050310467250.

[18] T. Mladenova, “A project management system for time planning and resources allocation,” *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2019 - Proceedings*, pp. 1299–1303, May 2019, doi: 10.23919/MIPRO.2019.8756834.

[19] M. B. Boutros, C. El Hajj, D. Jawad, and G. Martínez Montes, “Diffusion of ERP in the Construction Industry: An ERP Modules Approach: Case Study of Developing Countries,” *Buildings 2024, Vol. 14, Page 3224*, vol. 14, no. 10, p. 3224, Oct. 2024, doi: 10.3390/BUILDINGS14103224.

[20] M. Nolastname, E. Fernando, S. P. Hendratno, K. Nolastname, and S. Wifasari, “Enterprise resource planning systems: The business backbone,” *ACM International Conference Proceeding Series*, pp. 43–48, Apr. 2021, doi: 10.1145/3466029.3466049;PAGE:STRING:ARTICLE/CHAPTER.

[21] V. C. Sugiarto, R. Sarno, and D. Sunaryono, “Sales forecasting using Holt-Winters in Enterprise Resource Planning at sales and distribution module,” *Proceedings of 2016 International Conference on Information and Communication Technology and Systems, ICTS 2016*, pp. 8–13, Apr. 2017, doi: 10.1109/ICTS.2016.7910264.

[22] Z. P. Matolcsy, P. Booth, and B. Wieder, “Economic benefits of enterprise resource planning systems: some empirical evidence,” *Accounting & Finance*, vol. 45, no. 3, pp. 439–456, Nov. 2005, doi: 10.1111/J.1467-629X.2005.00149.X.

[23] J. Liu, Q. Chen, and Y. Qiu, “The Design of ERP Intelligent Sales Management System,” *Frontiers in Artificial Intelligence and Applications*, vol. 331, pp. 413–422, 2020, doi: 10.3233/FAIA200720.

[24] J. G. Nestell and D. L. Olson, *Successful ERP systems: a guide for businesses and executives*, First edition. in Information systems collection. New York, New York (222 East 46th Street, New York, NY 10017): Business Expert Press, 2018.

[25] H. Zhang, “A Deep Learning Model for ERP Enterprise Financial Management System,” *Advances in Multimedia*, vol. 2022, 2022, doi: 10.1155/2022/5783139.

[26] A. Reddy Kunduru, “Effective Usage of Artificial Intelligence in Enterprise Resource Planning Applications,” *International Journal of Computer Trends and Technology*, vol. 71, pp. 73–80, 2023, doi: 10.14445/22312803/IJCTT-V71I4P109.

[27] M. Kim, J. Jeong, and S. Bae, “Demand forecasting based on machine learning for mass customization in smart manufacturing,” *ACM International Conference Proceeding Series*, pp. 6–11, Apr. 2019, doi: 10.1145/3335656.3335658.

[28] M. S. Javaid, R. Chauhdary, A. Waleed, F. Ahmad, M. Zubair, and O. Tariq, “AI-Powered Smart Inventory Management: Enhancing Efficiency Through Predictive Analytics and Automation,” *2nd International Conference on Emerging Technologies in Electronics, Computing and Communication, ICETECC 2025*, 2025, doi: 10.1109/ICETECC65365.2025.11070285.

[29] O. Koca, O. T. Kaymakci, and M. Mercimek, “Advanced Predictive Maintenance with Machine Learning Failure Estimation in Industrial Packaging Robots,” *2020 15th International Conference on Development and Application Systems, DAS 2020 - Proceedings*, pp. 1–6, May 2020, doi: 10.1109/DAS49615.2020.9108913.

[30] M. A. Zaghdoudi, S. Hajri-Gabouj, C. Varnier, N. Zerhouni, and F. Ghezail, “Predictive Analysis Methodology for Industrial Systems: Application in Supplier Delays Prediction,” *2022 IEEE Information Technologies and Smart Industrial Systems, ITSIS 2022*, 2022, doi: 10.1109/ITSIS56166.2022.10118391.

[31] A. Nguyen *et al.*, “System Design for a Data-Driven and Explainable Customer Sentiment Monitor Using IoT and Enterprise Data,” *IEEE Access*, vol. 9, pp. 117140–117152, 2021, doi: 10.1109/ACCESS.2021.3106791.

[32] L. Sen Zhang, “Deep Learning-Based Optimization of Cloud Enterprise Resource Planning (ERP) Systems for Adaptive Decision Support and Management Effectiveness Analysis,” *IEEE Access*, vol. 12, pp. 193402–193415, 2024, doi: 10.1109/ACCESS.2024.3514879.

[33] S. Pungky and J. Wiratama, “Integrating Machine Learning Based Sales Forecasting With Odoo Erp for Automated Inventory Management in a Retail Company,” *Proceedings - 2025 4th International Conference on Electronics Representation and Algorithm: Artificial Intelligence: Creating Tomorrow’s World Today, ICERA 2025*, pp. 665–670, 2025, doi: 10.1109/ICERA66156.2025.11087334.

[34] I. Volkov and Y. Andreev, “Application of Machine Learning Methods for Production Data Analysis,” *Proceedings - 2025 International Russian Smart Industry Conference, SmartIndustryCon 2025*, pp. 984–989, 2025, doi: 10.1109/SMARTINDUSTRYCON65166.2025.10986132.

[35] C. Atik, A. Kut, D. Birant, and S. Birol, “Prediction of Cloth Waste Using Machine Learning Methods in the Textile Industry,” *2022 9th International Conference on Electrical and Electronics Engineering, ICEEE 2022*, pp. 165–169, 2022, doi: 10.1109/ICEEE55327.2022.9772517.

[36] S. Abaddi, “A proposed electric/hybrid batteries recycling hub with ERP implementation and simulation,” *Journal of Simulation*, Jul. 2025, doi: 10.1080/17477778.2025.2534661.

[37] Z. N. Jawad and Dr. V. B. János, “‘A comprehensive review of AI-enhanced decision making: An empirical analysis for optimizing medication market business,’” *Machine Learning with Applications*, vol. 20, p. 100676, Jun. 2025, doi: 10.1016/J.MLWA.2025.100676.

[38] H. J. Wahedi, M. Heltoft, G. J. Christophersen, T. Severinsen, S. Saha, and I. E. Nielsen, “Forecasting and Inventory Planning: An Empirical Investigation of Classical and Machine Learning Approaches for Svanehøj’s Future Software Consolidation,” *Applied Sciences 2023, Vol. 13, Page 8581*, vol. 13, no. 15, p. 8581, Jul. 2023, doi: 10.3390/APP13158581.

[39] C. J. de M. Santos, A. S. Barbosa, and A. M. O. Sant’Anna, “Machine Learning-integrated digital twins for process optimization in Industry 5.0,” *J. Ind. Inf. Integr.*, vol. 47, p. 100920, Sep. 2025, doi: 10.1016/J.JII.2025.100920.

[40] A. Metin and T. T. Bilgin, “Automated machine learning for fabric quality prediction: a comparative analysis,” *PeerJ Comput. Sci.*, vol. 10, p. e2188, Jul. 2024, doi: 10.7717/PEERJ-CS.2188/SUPP-1.

[41] J. Bender and J. Ovtcharova, “Prototyping Machine-Learning-Supported Lead Time Prediction Using AutoML,” *Procedia Comput. Sci.*, vol. 180, pp. 649–655, Jan. 2021, doi: 10.1016/J.PROCS.2021.01.287.

[42] “What is SAP ERP?” Accessed: Feb. 06, 2025. [Online]. Available: https://www.sap.com/products/erp/what-is-sap-erp.html

[43] J. Bernett *et al.*, “Guiding questions to avoid data leakage in biological machine learning applications,” *Nature Methods 2024 21:8*, vol. 21, no. 8, pp. 1444–1453, Aug. 2024, doi: 10.1038/s41592-024-02362-y.

[44] M. W. Mudiyanselage, H. Wu, and A. Mehrab, “Robust Day-Ahead Short-Term Energy Forecasting Using Cyclical Encoding and Attention-Driven Recurrent Networks,” *2025 Energy Conversion Congress and Expo Europe, ECCE Europe 2025 - Proceedings*, 2025, doi: 10.1109/ECCE-EUROPE62795.2025.11238539.

[45] “Tilastokeskus.” Accessed: Apr. 09, 2026. [Online]. Available: https://stat.fi/fi

[46] S. Onishi and S. Meguro, “Rethinking Data Augmentation for Tabular Data in Deep Learning,” May 2023, Accessed: Apr. 25, 2026. [Online]. Available: https://arxiv.org/abs/2305.10308v2

[47] D. Chicco, M. J. Warrens, and G. Jurman, “The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation,” *PeerJ Comput. Sci.*, vol. 7, pp. 1–24, Jul. 2021, doi: 10.7717/PEERJ-CS.623/SUPP-1.