2015

Forecasting obsolescence risk and product lifecycle with machine learning

Connor Patrick Jennings
Iowa State University

Forecasting obsolescence risk and product lifecycle with machine learning

by

Connor Patrick Jennings

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Industrial Engineering

Program of Study Committee:
Janis Terpenny, Major Professor
Mingyi Hong
Heike Hofmann

Iowa State University

Ames, Iowa

2015

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

ORML	Obsolescence risk forecasting using machine learning
LCML	Lifecycle forecasting using machine learning
RF	Random Forest
SVM	Support Vector Machine
ANN	Artificial Neural Networks
MSE	Mean Square Error

ABSTRACT

Rapid changes in technology have led to an increasingly fast pace of product introductions. New components offering added functionality, improved performance and quality are routinely available to a growing number of industry sectors (e.g., electronics, automotive, and defense industries). For long-life systems such as planes, ships, nuclear power plants, and more, these rapid changes help sustain the useful life, but at the same time, present significant challenges associated with managing change. Obsolescence of components and/or subsystems can be technical, functional, related to style, etc., and occur in nearly any industry. Over the years, many approaches for forecasting obsolescence have been developed. Inputs to such methods have been based on manual inputs and best estimates from product planners, or have been based on market analysis of parts, components, or assemblies that have been identified as higher risk for obsolescence on bill of materials. Gathering inputs required for forecasting is often subjective and laborious, causing inconsistences in predictions. To address this issue, the objective of this research is to develop a new framework and methodology capable of identifying and forecasting obsolescence with a high degree of accuracy while minimizing maintenance and upkeep. To accomplish this objective, current obsolescence forecasting methods were categorized by output type and assessed in terms of pros and cons. A machine learning methodology capable of predicting obsolescence risk level and estimating the date of obsolescence was developed. The machine learning methodology is used to classify parts as active (in production) or obsolete (discontinued) and can be used during the design stage to guide part selection. Estimates of the date parts will cease production can be used to more efficiently time redesigns of multiple obsolete parts from a product or system. A case study of the cell phone market is presented to demonstrate how the methodology can forecast product obsolescence

with a high degree of accuracy. For example, results of obsolescence forecasting in the case study predict parts as active or obsolete with a 98.3% accuracy and regularly predicts obsolescence dates within a few months.

CHAPTER I: INTRODUCTION

Obsolescence occurs in almost all industry sectors, generally due to the availability of alternatives that are more cost effective or can achieve better performance and quality, or a combination of both. Currently, 3% of the world's electronics products become obsolete monthly due to technical, functional, and style obsolescence [1]. For example, technical obsolescence occurs in the music industry. Music was first recorded to vinyls, then made portable by cassettes. Since the 1980s, compact discs have superseded cassettes. Recently, the music industry is observing technology shift from MP3 to music streaming services. Each societal shift causes immense amounts of obsolete inventory from audio players to physical music vessels. Functional obsolescence happens due to the lack of support for products, components or software even when the item can complete some form of the original task for which it was created. An example of functional obsolescence is a telegraph key which was used to transmit Morse code. Most telegraph keys are still fully functional and can still create the dots and dashes used to broadcast out messages. However, the lack of infrastructure connecting the telegraph keys has rendered them functionally obsolete. In addition, functional obsolescence commonly occurs in the software industry. For instance, organizations build information and communication technology (ICT) systems based on specific software packages. When software providers stop maintaining the software, organizations are faced with decisions such as whether it is best to update operating systems, not update and lose the entire system, or to keep the aging operating system and suffer lagging behind in applications and the risk of exposure through unpatched security holes. Obsolescence can also occur in the fashion industry where seasonal style changes can cause designs to suffer from style obsolescence. Moreover, obsolescence can even happen to human languages. For example, the

English word, “camelopard,” was adopted from Latin and Greek during 1750 to 1800 and has since been rendered obsolete by the more dominantly used word with the same meaning, giraffe.

Over the past few years, the flow of electronic components and software into traditionally non-electronic products have increased the problem of component and software obsolescence in more industries. Currently, most firms do not have effective and practical methods for predicting obsolescence risk and lifecycle. Because of this, firms are prone to over utilize reactive strategies [2]–[7]. Unfortunately, reactive strategies are often more costly than proactive strategies. Reactive strategies require additional resources (time and materials) to solve and can contribute to further delays that impact customer satisfaction. Proactive strategies allow firms to have more time to plan and react with an effective and low cost approach [2]–[5], [8]. Presently, the design and manufacturing sector lacks an obsolescence forecasting framework that can effectively predict product obsolescence while remaining easy to maintain by the organizations.

In this research, two machine learning-based methodologies that address obsolescence risk and lifecycle forecasting are presented. Specifically, one method is to address Obsolescence Risk Forecasting; the other method is to address Lifecycle Forecasting. Obsolescence risk forecasting and lifecycle forecasting are both umbrella terms under obsolescence forecasting. However, obsolescence risk forecasting refers to a process that predicts the probability that a given part will become obsolete. Lifecycle forecasting refers to a process that predicts the length of time during which the product will be in production. Both approaches can be adapted to forecast obsolescence in any scenario where obsolescence is present. The two techniques integrate machine learning to adapt over time to make the forecasts more accurate as more obsolete instances are observed by the model. Specifically, the objective of this research is to answer the following questions:

- How can large-scale product obsolescence forecasting be addressed using machine learning?
- Does machine learning based obsolescence forecasting improve on current obsolescence forecasting methods?

The contribution of this research is to introduce an effective approach for large-scale obsolescence forecasting using machine learning. To demonstrate the approach, a real-world application example is presented using three machine learning algorithms. Specifically, these machine learning algorithms are applied into a large data set of over 7,000 unique cell phone models with known in-production or out-of-production statuses.

The remainder of this thesis is organized as follows: In section 2, a brief overview of how obsolescence is handled in industry is presented, including: (1) current life-cycle forecasting methods, (2) current obsolescence risk forecasting methods, and (3) difficulties experienced in industry. In section 3, a brief overview of machine learning is presented. In section 4, the methodologies of Life Cycle Forecasting using Machine Learning (LCML) and Obsolescence Risk Forecasting using Machine Learning (ORML) are presented. Section 5 provides a case study of LCML and ORML that is used to predict obsolescence in the cell phone market. Section 6 discusses limitations of the LCML and ORML frameworks. Section 7 provides conclusions that include a discussion of research contribution and future work.

CHAPTER II: REVIEW OF LITERATURE

General Obsolescence

Obsolescence can have an immensely negative effect on many industries; the ramifications of which have generated a large body of research around obsolescence related decision making and more generally studying products through the product's life cycle. To address the economic aspect of obsolescence, cost minimization models are presented for both the product design side and the supply chain management side of obsolescence management [10]–[14]. Extensive work has also been conducted on the organization of obsolescence information [15], [15]–[17]. The organization of information allows one to make more accurate decisions during the design phase of a product lifecycle.

In practice, most firms do not have effective methods for predicting obsolescence and therefore are forced to over rely on reactive strategies [2]–[7]. The most common short term reactive obsolescence resolution strategies include lifetime buy, last-time buy, aftermarket sources, identification of alternative or substitute parts, emulated parts, and salvaged parts [15-16]. However, these strategies are only temporary and can fail if the organization runs out of ways to procure the required parts. More sustainable long-term alternatives are design-refresh and redesign. But these alternatives usually require large design projects and can carry costly budgets. Over time the cost of these reactive decisions add up. In a 2006 report, the U.S. Department of Defense (DoD) estimated cost of obsolescence and obsolescence mitigation for the government to be $10 billion annually for the U.S. government [9]. The estimates in the private sector could be higher because most small firms cannot afford the systems DoD uses to track and forecast obsolescence.

Obsolescence forecasting can be broken down into two groups, obsolescence risk forecasting and lifecycle forecasting. Obsolescence risk forecasting generates a probability that a part or other element may fall victim to obsolescence [18-21]. Life-cycle forecasting estimates the time from creation to obsolescence of the part or element [12, 16, 22, 23]. Using the creation date and life-cycle forecast, analysts can predict a date range for when a part or element will become obsolete [12, 16, 23, 24].

Obsolescence forecasting is important in both the design phase of the product and the manufacturing life-cycle of the product. It is estimated that 80-85% of cost during a product's life cycle are caused by decisions made in the design phase [27]. Understanding the risk level for each component in proposed bills of materials developed in the design phase, can help designers determine designs that have lower risk of component obsolescence and therefore reduce the life time cost impact. Additionally, obsolescence forecasting can be used throughout a product's life cycle to analyze predicted component obsolescence dates and find the optimal time to administer a product redesign that will remove the maximum number of obsolete or high obsolescence risk parts.

Life-Cycle Forecasting

The key benefit of lifecycle forecasting is that it allows analysts to predict a range of dates when the part will become obsolete [19]. These dates enable project managers to set timeframes in which they need to complete obsolescence mitigation projects, help designers understand when redesigns need to be accomplished and allow for inventory managers to more effectively manage inventory. All of these effects of lifecycle forecasting reduce the effect of obsolescence parts or elements in firms [19].

Currently, most lifecycle forecasting methods are developed based on the product life cycle model. The model is broken into six stages: introduction, growth, maturity, saturation, decline and phase-out. In Figure 1, the product life cycle model breaks down the x-axis into the six stages of time and the y-axis into the sales trends for each stage. After a successful product's sales fall enough to be considered in the phase-out range, the many firms will discontinue the product, which renders it unsupported and obsolete.

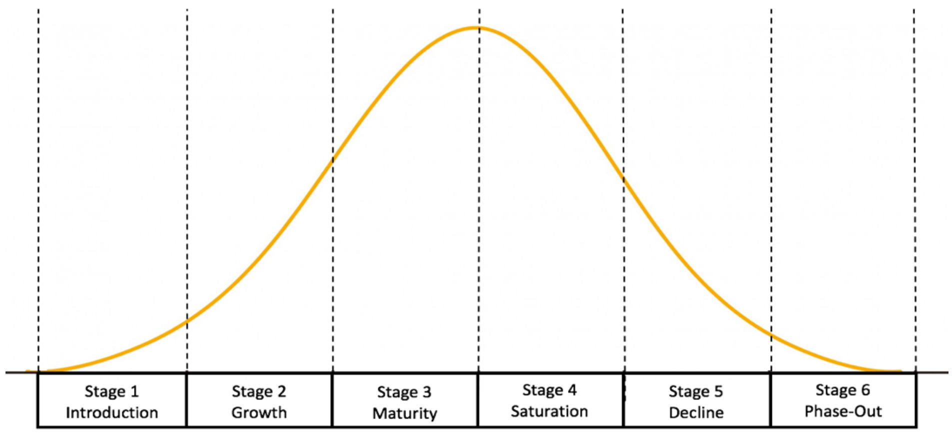

Figure 1: Product life cycle model [33]

Solomon et al. (2000) introduces the first obsolescence forecasting method that identifies characteristics to estimate the life stage of the product. These characteristics include indicators such as sales, price, usage, part modification, number of competitors and manufacturer profits [26]. The combination of these characteristics can then estimate the stage and whether or not the product is close to phase out. However, the lack of these signals only means the part will not go obsolete in the immediate future and these estimates are not useful for long-term predictions or if the part will become obsolete [19].

A current method for life-cycle forecasting is data mining sales data of parts or other elements and fitting a Gaussian (normal) trend curve to predict future sales over time [12, 23]. Using the predicted sales trend curve for a part, the peak sales is estimated by the mean (denoted as μ in Figure 2) and the stages are estimated as the standard deviations (denoted as σ in Figure 2) from the mean. When predicting obsolescence, the important area is the zone of obsolescence. This zone is given between +2.5σ to +3.5σ and gives the lower and upper bound time intervals for when a part or element will become obsolete [26].

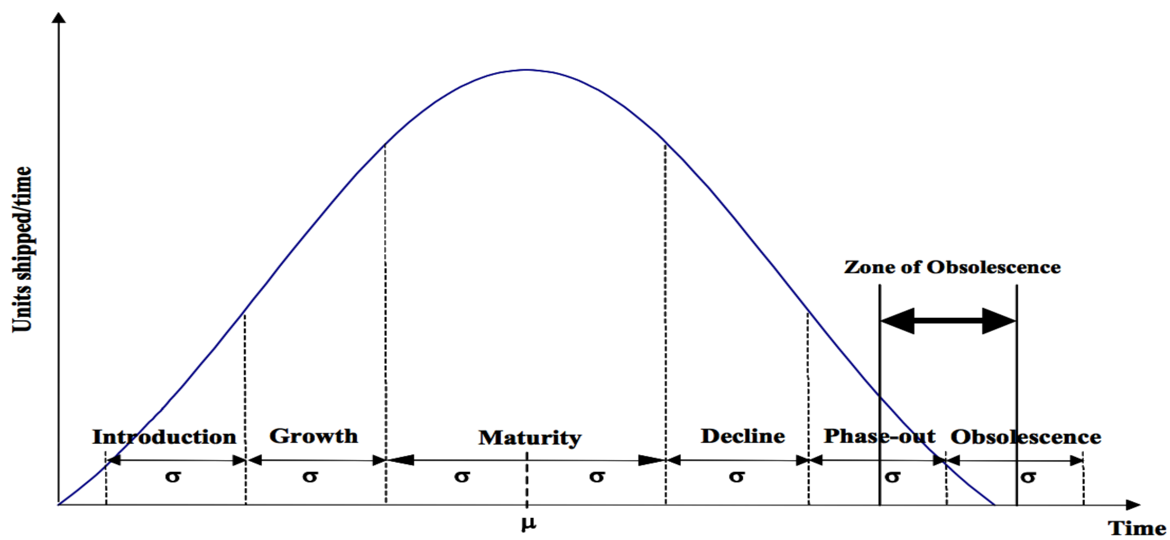

Figure 2 Life-cycle forecasting using Gaussian trend curve [26]

However, the assumption of normality of the sales cycle does not always hold true [28]. Another method involves organizing part information sales, price, usage, part modification, number of competitors and manufacturer profits into an ontology to better estimate the product life cycle stage the part is in, then fit a trend line using current sales to predict future sales [22, 24]. The zone of obsolescence is estimated using the predicted future sales, but does not assume normality because the factors outlined in [26] are used to estimate the stage, not the curve shape.

Currently, few life cycle forecasting methods in the literature do not use the concept of product life-cycle model. This method involves data mining parts information databases for introduction dates and procurement lifetimes to create a function with the input being the introduction date and the output being the estimated lifecycle [19]. The advantage of this method is the lack of reliance on sales data, the ability to create confidence limits on predictions and the simplicity of a model with one input and one output [19].

However, this model does not take into account each individual part's specifications. As a result, the model might be skewed. For example, two manufacturers with two different design styles both make similar products. The first manufacturer creates a well-designed product and predicts that the specifications will hold in the market for five years. The second manufacturer does not conduct market research and introduces a new product every year to keep specifications up to market standards. Over the next five years, the first company will have one long life data point and the second company will have five short life data points, this will skew the model into predicting the approximate lifecycle is shorter than it actually is because the model does not take into account specifications.

Obsolescence Risk Forecasting

Another common method used for predicting obsolescence is obsolescence risk forecasting. Obsolescence risk forecasting involves creating a scale to indicate the levels of the chance of a part or element becoming obsolete. The most common of these scales is to use probability of obsolescence [20, 21]. These scales, like product life-cycle stage predict, use a combination of key characteristics to identify where the part falls on the scale.

Currently, two simple models exist for obsolescence risk forecasting; both use high, medium, and low ratings for key obsolescence factors that can identify the risk level of a part becoming obsolete [18, 19, 21]. Rojo conducted a survey of current obsolescence analysts and created an obsolescence risk forecasting best practice that looks at numbers of manufacturers, years to end of life, stock available vs. consumption rate, and operational impact criticality as key indicator for potential parts with high obsolescence risk [23]. Josias and Terpenny also created a risk index to measure obsolescence risk [18]. The key metrics identified in this technique are manufacturers' market share, number of manufacturers, life cycle stage, and company's risk level [18, 19]. The weights for each metric can be altered based on changes from industry to industry. However, this output metric is not a percentage, but rather a scale from zero to three (zero being no risk of obsolescence and three being high risk).

Another approach introduced by van Jaarsveld uses demand data to estimate the risk of obsolescence. The method manually groups similar parts and watches the demand over time [22]. A formula is given to measure how a drop in demand increases the risk of obsolescence [22]. However, this method cannot predict very far into the future because it does not attempt to forecast out demand, which causes the obsolescence risk to be reactive [22].

Obsolescence Forecasting Scalability

Currently, advanced obsolescence forecasting frameworks have not been adopted by industry because of their inability to be implemented on a large scale. For a framework to be scalable, the framework must have the ability to adjust the capacity of predictions with minimal cost in minimal time over a large capacity range [29]. The frameworks discussed in the last two sections either have requirements for the model that does not work when scaled to industry needs or the model was so simple that bias can skew predictions. To achieve scalability in industry, obsolescence forecasting methods must meet the following requirements:

1. Do not require frequent (quarterly or more often) collection of data for all parts.

The reason for this requirement is that many methods involve tracking sales data of products to estimate where the product is in the sales cycle [12, 20, 23, 24]. A relatively small bill of material with 1000 parts would require a worker to find quarterly sales for 1000 parts and input them every quarter (or even more frequently). For large companies with numerous large bills of materials, any method requiring this becomes un-scalable.

2. Remove all human bias about markets.

The reason for this requirement is very similar to requirement 1; asking humans to input their opinion on every part quickly makes methods unrealistic for industry. Additionally, finding and interviewing subject matter experts for long periods of time can be a large cost and the experts' time would mostly be better spent doing another task. Also, subject matter experts may be biased in estimating the obsolescence risk of their field of expertise. The bias of these experts is largely due to the experts being so ingrained in the traditions of their field that new products or skills can seem inferior when in fact they will supersede the expert's aging traditions.

3. Account for multi-feature products in the obsolescence forecasting methodology.

Many methods have been developed to predict obsolescence of single feature products [12, 16, 23], for example flash drives. The flash drive may vary slightly in size and color but only has one key feature, memory. When a flash drive does not have sufficient memory to compete in the flash drive market, companies phase out that memory size in preference for ones with larger memory. Creating models for single-feature products like memory is easy because the part has only one variable that only causes one type of obsolescence, technical. However, multi-feature products, for example a car, can have many causes for becoming obsolete and this makes it much more challenging to model. Some examples are the style obsolescence of switching to not include cigarette lighters and ashtrays in cars and the removal of wood paneling on the sides of cars, the functional obsolescence of cassette, and now even CD, players for MP3 ports or Bluetooth, and the technical obsolescence of drum brakes giving way to the safer and longer running disc brakes. With these multiple obsolescence factors, many of the current forecasting models fall apart.

Table 1 List of all methodologies and scalability factors

Methods	Life-Cycle Forecasting	Obsolescence Risk Forecasting	Sales Data Required	Human Inputs	Multi-Feature Capable
Solomon et al. (2000)	✓	-	✓	✓	✓
Sandborn et al. (2005)**	✓	-	✓	-	-
Josias et al. (2009)	-	✓	-	✓	✓
van Jaarsveld et al. (2010)	-	✓	✓	✓*	✓
Sandborn (2011)**	✓	-	-	-	✓
Rojo et al. (2012)	-	✓	-	✓*	✓
Zheng et al. (2012)	✓	-	✓	✓	✓

Notes:
*Human bias due to manually filtering the BOM
**Sandborn 2005 & 2011 are different methods but the same creator

Any obsolescence forecasting method that does not meet these three requirements will most likely develop problems when trying to scale to meet the needs of industry. Table 1 shows the breakdown of obsolescence forecasting methods developed in the last 15 years, the type of obsolescence forecasting and if the methods meet the scalability factors. Ideally, a method would not require sales data or human input and be capable of multi-feature products. The only current method that meet these requirements is Sandborn (2011) [19]. Even though the simplicity of the Sandborn (2011) [19] method allows it to be scalable, the problem of many short life product data points pulling down life cycle estimates of manufactures of similar products with longer lives still stands. The introduction of a new obsolescence forecasting approach is necessary because of the lack of scalability and accuracy in any current method.

Machine Learning

The goal of machine learning is to build computer systems that automatically improve with additional information [30]. The continual improvement of a system makes machine learning a perfect method for the prediction of obsolescence risk and forecasting of product life cycles because over time, more parts become obsolete in markets and this creates additional information from which the system can learn and improve.

Machine learning has gained popularity in many application fields because it can process large data sets with many variables. The applications of machine learning range from creating better recommendation systems on Netflix to facial recognition in pictures to cancer prediction and prognosis [28–30]. Specifically, in the field of design, machine learning has been used to gather information and develop conclusions from previously under utilized sources. For example, public online customer reviews of products are mined to better understand how customers feel about individual product features [34]. The results of these analysis can be used to improve products during redesign and new product development by understanding customer's true desires in products. Another example of data mining and machine learning in design is the analysis of social media for feedback on products. Current work has showed that by using social media data, machine learning can predict sales of product and levels of market adoption [35]. Understanding the market adoption of features can indicate if the feature is a passing or a permanent trend.

Another application of machine learning was conducted in France where researchers measured the effectiveness of search results on a newspaper's website. The researchers used machine learning to show how manually preset search grouping methods could become obsolete over time [36]. For example, the abbreviation CDC could stand for 'Caisse des Dépôts et Consignations' but if there is a disease outbreak, CDC as a search term on a news website could change meanings to 'Center of Disease Control' [36]. The machine-learning algorithm was able to notify the newspaper of these obsolete rules and suggest changes. An overlooked contribution of the study was the first and only utilization of machine learning to classify an object or element as current or obsolete. The study shows that machine learning is an acceptable technique for analyzing obsolescence. However, the study does not give details on how to have a generalizable framework for forecasting obsolescence.

In all, product obsolescence is a complex and costly problem in industry. A shift to more proactive obsolescence mitigation strategies would greatly increase the potential options while decreasing the cost impact. Current obsolescence forecasting methods use sales data, human input or are not capable of forecasting multi-feature products. These method requirements limit the ability of current approaches to be implemented and maintained by industry. Machine learning is commonly utilized in predictive analysis of many large scale industrial data-driven problems. The research presented in this thesis will introduce and demonstrate that machine learning is an efficient and accurate tool for forecasting obsolescence.

CHAPTER III: METHODOLOGY

Two separate, obsolescence forecasting methodologies and frameworks are introduced in this research. Both approaches apply machine learning to improve the adoptability of the method for industry. The two approaches are differentiated by the two major outputs of the model. The first outputs the risk level that a product or component will become obsolete and is called obsolescence risk forecasting using machine learning (ORML). The second method outputs an estimation of the date the product or component will become obsolete and is called life cycle forecasting using machine learning (LCML).

Both ORML and LCML use a subset of machine learning called supervised learning. Supervised learning creates predictive models based on data with known labels. These predictive models are used to predict labels of new and unknown data. A common introduction problem in supervised learning is to create a model to predict whether an individual will go outside or stay inside based on the weather. Two data sets are presented and follow the process shown in Figure 3. The first data set contains the temperature, humidity, and sunniness for each day and whether the subject stayed inside or went outside. This data set is the training data set because a predictive model with output, stay inside or go outside will be trained using this data. The training data set is feed into a machine-learning algorithm, which creates a model or rules that will most accurately classify the known label based on the known weather information. The algorithm produces a predictive model and weather information where the label is unknown can be run though the predictive model to get an expected label of whether the individual will go outside or stay inside. The unknown data set is also called the test set because it will be used to test the accuracy of the predictive model. For the stay inside or go outside prediction model and all supervised learning models, the more data with known labels submitted to the machine learning algorithm the more effective the predictive model. This means supervised machine learning is a strong fit for any problem where data continually flows in and can make the predictions more accurate. With prediction of product obsolescence, the stream of newly created and discontinued products allows the predictive models created using ORML and LCML to gain accuracy over time.

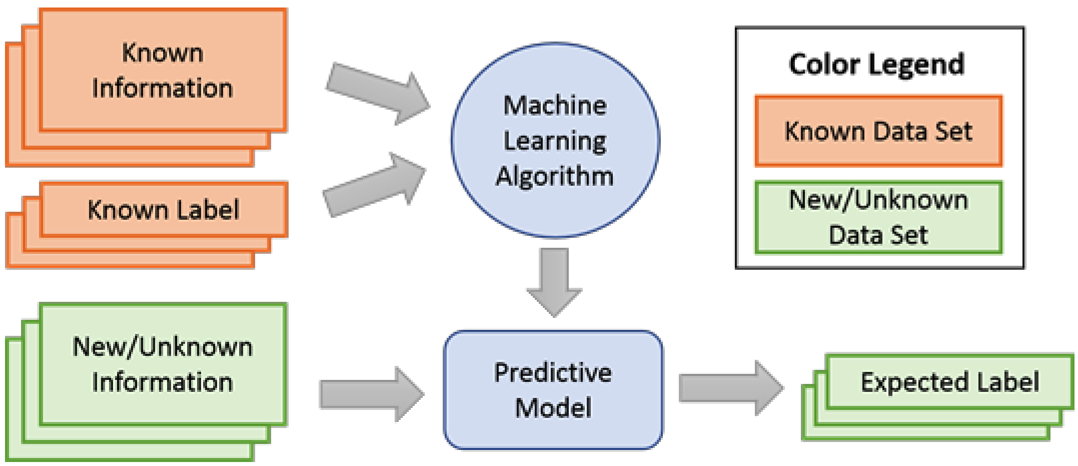

Figure 3 Supervised learning process

Supervised machine learning was chosen over unsupervised machine learning because unsupervised does not have a known data set. Unsupervised machine learning does not have a label to predict, but rather uses algorithms to fix clusters and patterns in the data. Similar methods could be advantageous to identifying groups of comparable products for product redesign or for cost reduction in the design phase. However, due to unsupervised machine learning finding groupings that are not explicitly obsolete vs. active, supervised learning was chosen over unsupervised learning for this obsolescence forecasting framework.

Additionally, machine learning models are not deterministic models. Many algorithms use randomization to split variables and evaluate the outcome. A byproduct of this trait is that the predictive models will vary slightly each time the algorithm is implemented. Even with these slight variations, machine learning models are highly effective and used in many predictive applications.

Obsolescence Risk Forecasting using Machine Learning

The forecasting methods introduced and demonstrated in this research are based on the concept that parts become obsolete because other products in the market have a superior combination of features, software, and/or other added value. The Obsolescence Risk Forecasting using Machine Learning (ORML) framework, much like the weather example, are shown information and attempt to classify the part with the correct label. However, instead of weather information, the technical specifications of current active and obsolete parts are fed into the algorithms to create the predictive models. In Figure 4, after the predictive model is created, the technical specifications of parts with unknown obsolescence statuses are structured in the same way as the known parts and input to the predictive model. The model outputs the probability that the part is classified with the label active or obsolete. The probability the part is obsolete can be used to show the obsolescence risk level. This risk level can be used in the design stage for part selection and in inventory management to help understand the risk of suppliers stopping production.

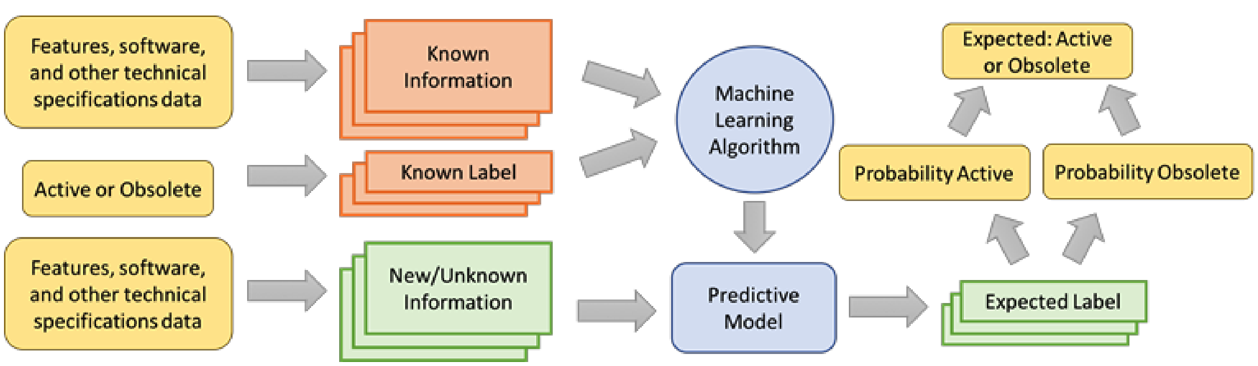

Figure 4 Obsolescence risk supervised learning process

Product A shows a product with a 100% chance of the part being active. Product B demonstrates a mix prediction between 60% chance of being active and 40% of being obsolete. Product C shows prediction of a product with a 100% chance of being obsolete.

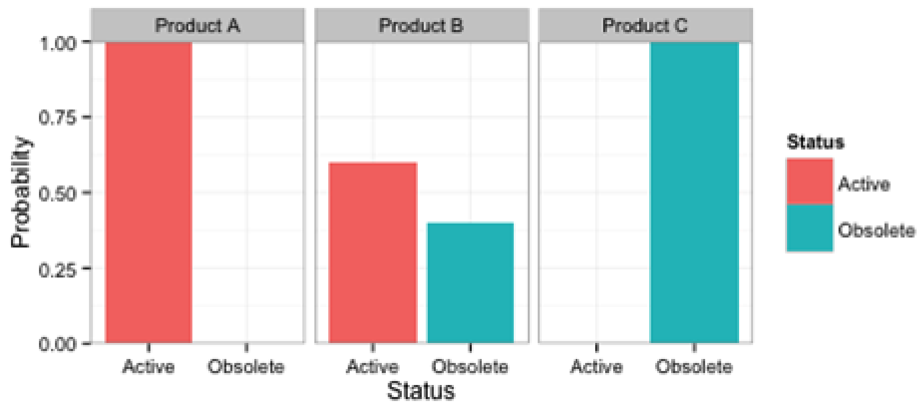

Figure 5: Outputs of ORML

Life Cycle Forecasting using Machine Learning

The LCML framework is built on the same principal that parts become obsolete because other products in the market have a superior combination of features, software, and/or other added value, the difference is what the frameworks are predicting. Where ORML predicts the label active or obsolete, LCML uses regression to predict a numeric value of when the product/component will stop being manufactured.

LCML's ability to estimate a date of obsolescence is a highly useful metric. LCML will give designers and supply chain professionals a more effective way of predicting the length of time to complete redesign or find a substitute supplier or component. Understanding when each component on a bill of materials will become obsolete will allow designers not only the ability to give time constraints on projects, but more effectively time redesign projects to maximize the number of high risk components removed from the assembly.

CHAPTER IV: CASE STUDY

The case study serves to demonstrate the accuracy and scalability of ORML and LCML as methods to forecast obsolescence. The data contains over 7000 unique models of cellular phones with known active in-production or discontinued status, release year and quarter, and other technical specifications. The specifications include weight (g.), screen size (inch.), screen resolution (pixels), talk time on one battery (min.), primary and secondary camera size (MP), type of web browser, and if the phone has the following: 3.5 mm headphone jack, Bluetooth, email, push email, radio, SMS, MMS, thread text messaging, GPS, vibration alerts, or a physical keyboard. The data was collected from one of the most popular cell phone forums, GSM Arena using a web scraper. The original dataset, and the code for the web scraper, and machine learning models created in this case study can be download at connorj.github.io/research. GSM Arena is an online forum that provides detailed and accurate information about mobile phones and their features. For this reason, the data set can have missing values and even miss reported information. Even with these short falls with the data set, this more accurately represents data collected in industry and demonstrates the robustness of the ORML and LCML frameworks.

The data set was formatted into a machine learning friendly format. The example data set is provided in Table 2. The specification and information are outlined in green and the labels are outlined in blue. The brand and name of a cell phone are used as the unique identifier of the phone.

Table 2 Sample of case study data

Brand	Phone	Availability		Release Yr.	Quarter	Screen Size	Talk Time	Weight	Camera	Browser	GPS
Apple	iPhone	Discontinued		2007	Q3	3.5	480	135	2	HTML	No
Apple	iPhone 3G	Discontinued		2008	Q3	3.5	600	133	2	HTML	Yes
Apple	iPhone 3GS	Discontinued		2009	Q3	3.5	720	135	3.15	HTML	Yes
Apple	iPhone 4	Available		2010	Q2	3.5	840	137	5	HTML5	Yes
Apple	iPhone 4s	Available		2011	Q4	3.5	840	140	8	HTML5	Yes
Apple	iPhone 5	Available		2012	Q3	4	480	112	8	HTML5	Yes
Apple	iPhone 5c	Available		2013	Q3	4	600	132	8	HTML5	Yes
Apple	iPhone 5s	Available		2013	Q3	4	600	112	8	HTML5	Yes
Apple	iPhone 6	Available		2014	Q3	4.7	840	129	8	HTML5	Yes
Apple	iPhone 6+	Available		2014	Q3	5.5	1440	172	8	HTML5	Yes

After formatting the data, the data set was split into two random groups. The first group represents ⅔ of the data set and is called the training data set. The training set is the data set used to create the prediction model. The second is the test set and represents the other ⅓. Although all the data sets are known in this case study, the test set will be put through the predictive model and accuracy will be determined by comparing actuals obsolescence statues and obsolescence date verses the one predicted by the model. This practice is known as cross-validation and is a best practice for model creation and evaluation because the data used to create a prediction model is never used to validate its accuracy [37]. Currently, the majority of the obsolescence forecasting models in the literature estimate model accuracy by using the same data used to create the model[19], [24], [26]. The data set was split into a 1/3 test set and a 2/3 training set for an initial analysis for accuracy using confusion matrixes. A more in-depth analysis was conducted where the ratio of training and test set sizes were changed and accuracy were accessed (Table 6 & 8).

The next step in the case study was to run the training data set through a machine-learning algorithm to create a predictive model. Machine learning has many algorithms and infinitely more if counting all the slight variations that can be done to increase accuracy. Three machine learning algorithms, artificial neural networks (ANN), support vector machines (SVM), and random forest (RF) will be applied for this case study [38]–[40]. Decision trees and support vector machines were ranked first and third, respectfully on the list of the "Top 10 Algorithms in Data Mining" [41]. However, standard decision trees are often inaccurate and over fit data sets [42]. Random forest, an aggregation of many decision trees, averages the trees with the intention of lowering the variance of the prediction [42]. For this reason, random forest was selected over standard decision trees. The algorithm listed second, K-means, is a unsupervised clustering method and would group similar products together rather than forecast an output. For this reason, K-means is not a possible alternative for algorithm to be used for either ORML and LCML and therefore was not included in this case study. Although artificial neural networks were not in the top 10, artificial neural networks were selected based on their use in detecting rule-based filtering models for newspapers topic search; making artificial neural networks the first machine learning algorithm used to identify obsolescence [36] .

The final step is once the algorithm constructs a predictive model, to have each part or element from the "unknown" data set run through the model and receive a predicted label.

Results for Obsolescence Risk Forecasting

The accuracy for the ORML model will be represented in a confusion matrix. The confusion matrix (Tables 3, 4, & 5) shows how many cell phones were classified correctly vs. incorrectly. Numbers in the (available, available) and (discontinued, discontinued) cells are correctly classified and all other cells are miss classified.

The first algorithm used was ANN. The neural networks classification was done in R 3.0.2 using the package “caret” [43]. The probability of each part being available or discontinued was outputted and the highest probability was assigned. The actual statuses were compared to the predicted values and a confusion matrix was developed (Table 3). The model correctly predicted 91.66% of cell phones in the test data set.

Table 3: Neural network ORML confusion matrix for cell phones

	Prediction			
		Available	Discontinued	Total
Actual	Available	1295	67	1362 (95.08%)
	Discontinued	129	860	989 (86.96%)
	Total	1424 (90.94%)	927 (92.77%)	2351 (91.66%)

The next algorithm applied was SVM. The support vector machine utilized the SVM classification function from the package “e1071” [44] in R 3.0.2. The algorithm was implemented on the training data set that contained 66.6% of the total data. The prediction model then classified the remaining 33.3% of phones not used in the model creation. The actual statuses and the predicted statuses were compared and the confusion matrix in Table 4 was created. The SVM model has a model accuracy of 92.4%.

Table 4: Support Vector Machine ORML confusion matrix for cell phones

	Prediction			
		Available	Discontinued	Total
Actual	Available	1218	76	1294 (94.13%)
	Discontinued	92	827	919 (89.99%)
	Total	1310 (92.98%)	903 (91.58%)	2213 (92.41%)

The last algorithm applied was RF. The model was implemented in R 3.0.2 using the package “randomForest” [45]. The model was trained with a 66.6% training set and was tested with 33.3%. The predicted test set and the actual statuses were compared in Table 5. The model received an accuracy of 92.56%. This was the higher of all three algorithms.

Table 5 Random forecast ORML confusion matrix for cell phones

	Prediction			
		Available	Discontinued	Total
Actual	Available	1243	72	1315 (94.52%)
	Discontinued	98	873	971 (89.91%)
	Total	1341 (92.69%)	945 (92.38%)	2286 (92.56%)

For Table 3, 4, & 5, the training size was held constant at 66.6%. Table 6 illustrates how changing the percent of instances from the data set used to create the model affects accuracy. ANN and SVM preform at about the same accuracy for every training size, while RF always performs at a higher accuracy.

Table 6: Average Accuracy of Predictions by Training Size for ORML

Training Size (%)	Random Forest			Neural Network			Support Vector Machine		
	Training (%)	Testing (%)	Overall (%)	Training (%)	Testing (%)	Overall (%)	Training (%)	Testing (%)	Overall (%)
50	98.8	92.2	95.5	91.8	91.2	91.5	90.9	91.7	91.3
60	98.5	92.5	96.1	91.4	91.7	91.5	91.0	92.2	91.4
70	98.5	92.9	96.8	91.5	91.9	91.6	91.3	92.3	91.6
80	98.2	93.3	97.2	91.7	91.1	91.6	91.6	91.7	91.6
90	98.2	94.3	97.8	91.7	91.2	91.6	91.7	91.2	91.6
100	-	-	98.3	-	-	91.1	-	-	91.6

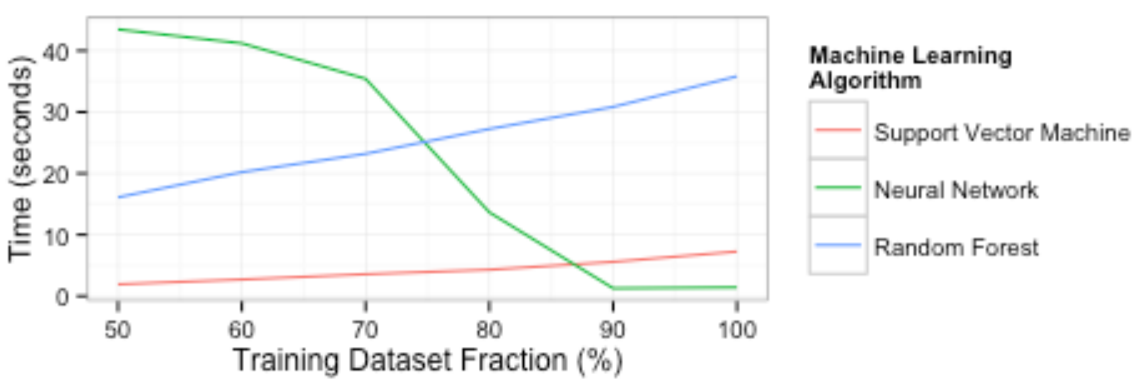

Figure 6: Overall average evaluation speed by training dataset fraction for ORML

The algorithms were compared using the prediction model creation time for each of the 50-100% training sets used in Table 5. Ten predictive models were created for each training size and the average time was plotted (Figure 11). Both SVM and RF increase in time at a near constant rate while ANN decrease in model creation time while the training set grows in size. SVM is the fast algorithm. Then both RF and ANN are slower which RF being slightly faster on average.

Table 7: Summary of model preference ranking for ORML

	RF	ANN	SVM
Performance based characteristics			
Accuracy	1st	3rd	2nd
Evaluation Speed	2nd	3rd	1st
Non-performance based characteristics			
Interpretability	1st	3rd	2nd
Maintainability/flexibility	1st	2nd	3rd

Four characteristics, identified in Zhang & Bivens 2007, were measured to rank the algorithms. The first two characteristics are performance based: accuracy and evaluation speed. The rankings of the algorithms in Table 7 for the first two attributes is done by best model accuracy and by average time to complete the ten simulations of each of the six different training set sizes. The second two characteristics are usability based: interpretability and maintainability/flexibility. Interpretability is defined as the ability for analyst to comprehend the model and analyze the output. Maintainability/flexibility represents the models ability to adapt over time and how much work is required to keep the model running.

Random Forest was ranked number one in interpretability due to the visual nature of decision trees and the ability for analyst to follow the flow of the tree to understand the steps in the classification model. Support Vector Machine was ranked second because the concept of creating a plane to separate the available and discontinued groups is easy to understand, but because of the high dimensionality of the data there is no obvious visual representation of this model. Lastly, neural networks were ranked third out of three because of the complexity of the trained network and the 'black-boxness' of this classification method.

Maintaining machine-learning models requires regular inputs of data to maintain the accuracy of the model because both neural networks and support vector machines require only numeric variables, all variables must be converted to numeric. Creating numeric indexes can be time consuming and will slow down the data entry process, for this reason random forest was ranked number one. In Table 6, as the training set size decrease, the accuracy of the neural network test set dropped faster than the support vector machine test set. While fewer data points the neural network was not flexible and could not preform as well as the support vector machine because of this support vector machine was ranked second and neural networks third.

Overall, random forest was ranked first in all attributes besides speed where it was ranked second. For this reason, random forest is the most appropriate algorithm for ORML in the cell phone market. This result can be verified by the accuracy of the random forest model were 100% of the data set is used to create and test the model. Random forest was able to correctly classify 98.3% of the cellphones.

Results for Life Cycle Forecasting

The following section contains the results of the cell phone case study to forecast obsolescence by using the LCML framework. First, the results of the 2/3 training set and 1/3 test set are shown and discussed. Similar to the ORML section, the model accuracy is examined as the training size changes and the speed of each algorithm is accessed. Finally, each algorithm is ranked based on the four characteristics, accuracy, evaluation speed, interpretability, and maintainability/flexibility.

The LCML framework predicts the date the product/component will become obsolete. Since the output is a numeric rather than a binary classifier, the results can not be easily presented in a confusion matrix. For this reason, the actual obsolescence dates vs. the predicted obsolescence dates was plotted to visually represent the accuracy of each model. A red dashed line at 45 degrees was plot to show a prefect 1 to 1 prediction rate. Unlike ORML, to access the model accuracy, the percentage correct can not be used as the gauge of model success. For the LCML framework mean square error (MSE) will be used to determine accuracy. The equation for MSE is as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

Where n equal to the number of predictions made, $\hat{Y}$ is the predicted obsolescence date, and Y is the actual obsolescence date. The lower the MSE, means the predicted and actual values are closer, therefore lower MSE means the model has a higher accuracy.

One large challenge of the LCML section of the case study was the lack of obsolescence dates available through our web scraping data source. Users of the cell phone web forum commonly updated cell phone specification and whether the phone was in production or discontinued, but rarely listed an explicitly date of obsolescence. For this reason, substantially less data was available for the LCML case study.

The first algorithm tested with the LCML framework was ANN. The neural networks require a large amount of data to create accurate prediction models. Since the LCML data set was smaller, the neural network was unable to create a model. If no model is created, then the algorithm defaults to taking an average of the training set and always applying the average for all predictions. The results of this method are shown in Figure 9. The prediction model received a MSE of 4.77. The square root of the MSE will find the average prediction error and for neural networks the average prediction had an error of 2.18 years. An error that large would not be effective in a market when the average life span of a product is only 1-2 years.

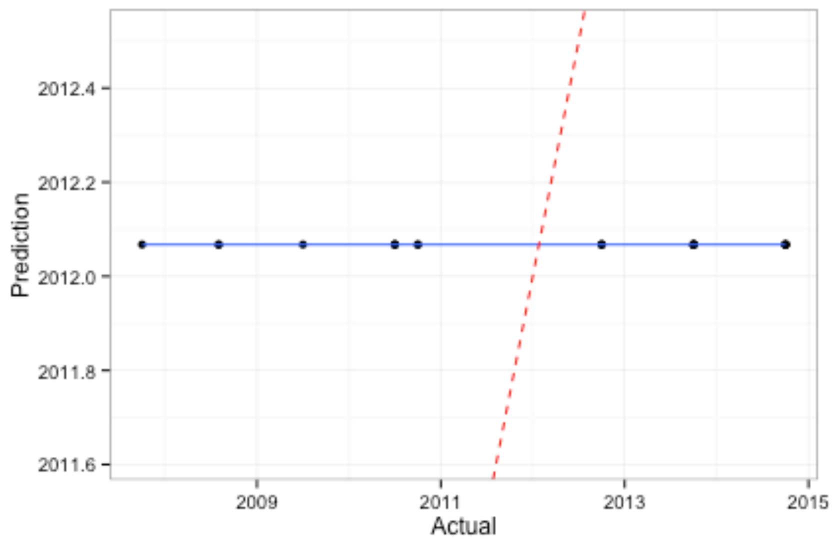

Figure 7: Actual vs. Predicted End of Life using Neural Networks and LCML

The next algorithm applied was SVM. In contrast to neural networks, SVM utilized the smaller data set and created an accurate prediction model (Figure 10). In Figure 10, the blue is a line of best fit of the actual vs. predicted end of life. The best fit line and the red “prefect prediction” line are fairly similar. The MSE of the model is 0.36 and is much accurate than the MSE of 4.77 for neural networks.

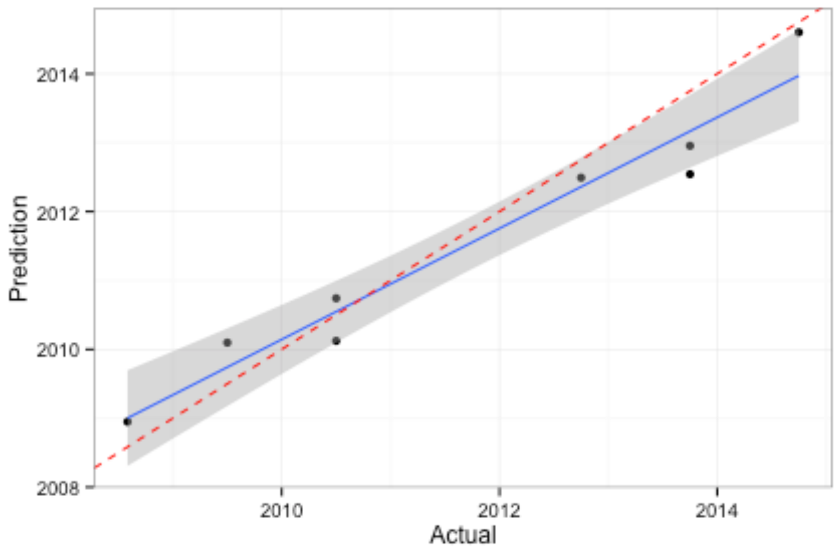

Figure 8: Actual vs. Predicted End of Life using SVM and LCML

The last algorithm testing the LCML framework was RF. Random forest, similar to SVM, constructed an accurate obsolescence date prediction model. The model has a 0.52 MSE. The slightly higher model error rate can be seen when comparing Figure 10 and Figure 11. SVM was capable of predicting closer to the red dashed or “prefect prediction” line.

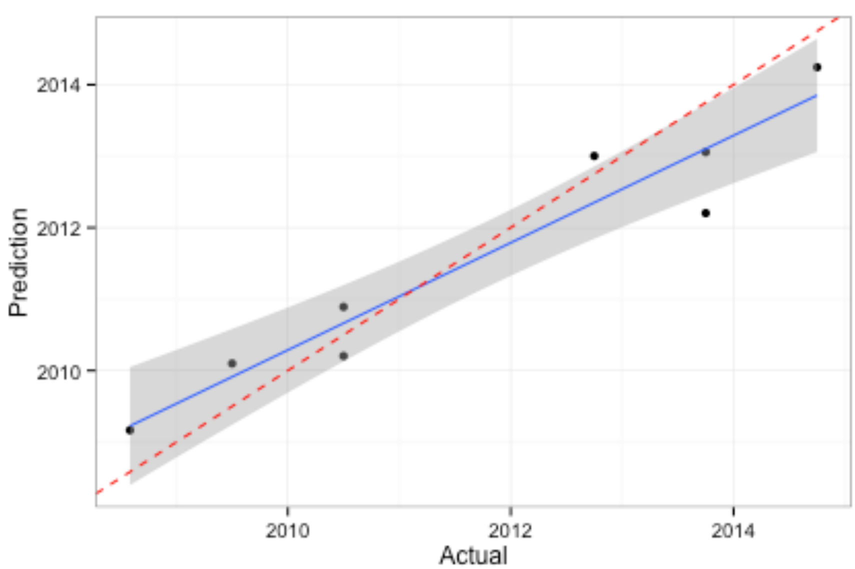

Figure 9: Actual vs. Predicted End of Life using Random Forest and LCML

Table 8: Average MSE of Predictions by Training Size for LCML

Training Size (%)	Random Forest			Neural Network			Support Vector Machine		
	Training	Testing	Overall	Training	Testing	Overall	Training	Testing	Overall
50	0.47	2.00	1.27	4.71	5.73	5.10	0.36	0.88	0.56
60	0.41	1.81	1.01	4.75	5.67	5.10	0.33	1.41	0.74
70	0.40	1.22	0.68	4.70	5.89	5.15	0.34	1.02	0.60
80	0.39	0.74	0.48	4.80	5.65	5.12	0.39	0.92	0.59
90	0.33	1.09	0.44	4.87	5.75	5.21	0.32	1.34	0.71
100	-	-	0.36	-	-	5.21	-	-	0.60

An analysis of how changing the training set size effects the prediction model's accuracy was conducted. The model was created with the training set and then tested on the training, testing, and overall data set. Each was conducted ten times and the MSE was averaged. For neural networks, the MSE remained constant through out training size changes. This was largely due to the model only using the average obsolescence date to predict the obsolescence dates in other predictions. Random forest was a steady decrease in model error as the training sizes increased, while SVM had a more constantly low model error.

The times to create each model was recorded and plotted in Figure 12. Neural networks took nearly no time to average the dates in the training set. SVM was slightly slower than neural networks, but forecasted the obsolescence date with a far greater accuracy. Random forest was third and was almost 8 times slower than SVM.

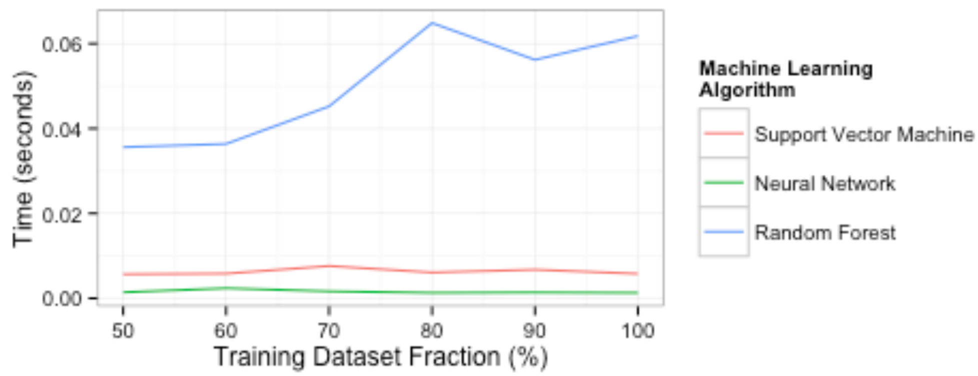

Figure 10: Overall average evaluation speed by training dataset fraction for LCML

Table 9: Summary of model preference ranking for LCML

	RF	ANN	SVM
Performance based characteristics			
Accuracy	2nd	3rd	1st
Evaluation Speed	3rd	1st	2nd
Non-performance based characteristics			
Interpretability	1st	3rd	2nd
Maintainability/flexibility	1st	3rd	2nd

The last step in the algorithm analysis was to rank the algorithms by the four key characteristics outline previously in this paper. Although random forest was rated higher in both non-performance based characteristics, SVM preformed much better on accuracy and speed. For these reasons, support vector machines are the most appropriate algorithm for forecasting obsolescence dates using LCML in the cell phone market.

CHAPTER V: LIMITATIONS

Like all the obsolescence forecasting frameworks, LCML and ORML have limitations and problems that may compromise the validity of the estimations. This section addresses these problems and limitations to provide users with a better understanding of the frameworks.

The first problems can arise from the first step, data collection. The data must have fairly reliable and up to date data. As demonstrated in the case study, the data does not need to be complete but the more complete the data is the more accurate the prediction. Another important part of the data formatting process is variable selection and creation. The correct variables can easily capture the change in the market and can indicate when parts or elements are becoming obsolete. However, these variables might not always be a simple measure of memory, screen resolutions or another metric. For example, a variable may need to be created to denote the highest, medium, and lowest memory levels of a phone. Apple, Inc. usually ends production of the highest and medium versions of a phone, but still produces the lowest memory version of the prior model phone to capture the market of people looking for a “cheap” iPhone. The size of memory in the lowest memory version of the iPhone has changed over time and using only phone memory would not capture this trend in the predictive model.

With the diversity of industry where obsolescence is present and these frameworks can be used, there will be no uniform indicator between industries. A good metric to measure obsolescence for flash drives is probably memory, however for cell phones the features like thread text messaging and screen resolution are more useful than memory. Furthermore, good metrics can change over time. When cellphones were first invented, connectivity was one of the most important factors and little emphasis was on features. Now connectivity is a given and features determine phone obsolescence.

Another problem with obsolescence forecasting frameworks is finding acceptable prediction accuracies from industry to industry. An industry like transistors, with exponential change such as described by Moore's Law, would likely be predicted more accurately than the cell phone market due to the complexity of the products and different marketing and pricing aspects.

The last problem is one that plagues all machine learning and statistical models. If the data used to build the model does not represent the current real world, the model will not be effective. In obsolescence, there is an extremely high chance of this occurring due to rapid innovation or invention. When Apple released the first iPhone it was the first in many categories and because of that, it accelerated the obsolescence of many of the phones in the current market. A machine learning or statistical obsolescence model at the time built with past obsolescence data would not predict the jump in technology this innovation would cause. This means the obsolescence forecasting frameworks introduced in this research and all current obsolescence models cannot predict large jumps in innovation, but are better suited to track steady improvements in an industry.

CHAPTER VI: CONCLUSIONS AND FUTURE WORK

The case study demonstrated the power of the ORML by correctly identifying active and obsolete cell phone with an accuracy as high as 98.3%. Random forest was selected as the best algorithm for the ORML framework in the cellphone market based on model accuracy, speed, interpretability, and maintainability/flexibility. The second half of the case study, showed the accuracy of the LCML framework and showed that cell phones obsolescence dates can be predicted with in a few months of the actual obsolescence date. The best algorithm for LCML in the cell phone market was support vector machines based on the four key characteristics named above.

One of the contributions of this paper is introducing the two category types of obsolescence forecasting: obsolescence risk and life cycle. Each method was examined for its ability to scale using the three characteristics: requiring sales data for all products in each component's market, human inputs for each part, and has the capability to handle multi-feature products/components. Machine learning was introduced as a technique employed to utilize knowledge in large data sets and help automate complex systems. This made machine learning a prime candidate for solving the problem of scaling obsolescence forecasting models to industries' needs. The first machine learning framework introduced was Obsolescence Risk Forecasting using Machine Learning (ORML) and this outputted a risk index of each product being active or obsolete. The second machine learning framework was Life Cycle Forecasting using Machine Learning (LCML) and this framework outputted an estimate of the life span of the product. A case study using ORML and LCML was demonstrated using over 7000 cell phones and showed the high level of accuracy of these frameworks. Then the limitations of applying these frameworks to current obsolescence forecasting systems were discussed to better understand the implications and potential causes for inaccuracy.

With obsolescence effecting almost all industries, reducing the cost of impact would save millions of dollars annually. The easiest way to reduce the impact is by involving obsolescence mitigation planning in earlier phases of design and supply chain management. This shift from a reactionary approach to a proactive approach would only be possible through more accurate obsolescence forecasting that can scale to industries' needs. This research establishes machine learning as a capable technique to meet industries' large scale needs while maintaining an extremely high accuracy for predicting obsolescence.

The results and frameworks discussed in this research should spark businesses to implement their own case studies to better analyze obsolescence within their organization. These additional studies will show further that machine learning is a highly effective solution to obsolescence forecasting. Additionally, the creation of supplementary tools built to utilize obsolescence risk and life cycle predictions would help industry to transfer from reactive to proactive. Some examples of tools built on top of these prediction models could include software where bills of materials can be submitted and the risk levels of each component can be calculated. These individual risk levels could be combined to show overall risk levels of different designs. The overall risk levels can be used in the early design stage and even in redesigns to help choose the best design to minimize the impact of obsolescence through the product's life cycle. Life cycle forecasting could also be used to help make life-time buy or last buy orders more accurate by better understanding when the products will no longer be manufactured.

CHAPTER VII: REFERENCES

[1] QTEC, 2006.

[2] C. Pince and R. Dekker, “An Inventory Model for Slow Moving Items Subject to Obsolescence,” 2009.

[3] Y. Song and H. C. Lau, “A Periodic-Review Inventory Model with Application to the Continuous-Review Obsolescence Problem,” *Eur. J. Oper. Res.*, vol. 159, pp. 110–120, May 2003.

[4] E. Menipaz, “An Inventory Model with Product Obsolescence and its Implications for High Technology Industry,” *IEEE Trans. Reliab.*, vol. 35, no. 2, pp. 185–187, 1986.

[5] A. Persona, A. Grassi, and M. Cetena, “Consignment stock of inventories in the presence of obsolescence,” *Int. J. Prod. Res.*, vol. 43, no. 23, pp. 4969–4988, May 2005.

[6] R. Cordero, “Managing for Speed To Avoid Product Obsolescence: A Survey of Techniques,” *J. Prod. Innov. Manag.*, vol. 8, no. 4, pp. 283–294, Dec. 1991.

[7] F. J. Romero Rojo, R. Roy, E. Shehab, and P. J. Wardle, “Obsolescence Challenges for Product-Service Systems in Aerospace and Defence Industry,” presented at the CIRP Industrial Product-Service Systems Conference, 2009, p. 255.

[8] R. Rai and J. Terpenny, “Principles for Managing Technological Product Obsolescence,” *IEEE Trans. Compon. Packag. Technol.*, vol. 31, no. 4, pp. 880–889.

[9] A. Kleyner and P. Sandborn, “Minimizing life cycle cost by managing product reliability via validation plan and warranty return cost,” *Int. J. Prod. Econ.*, vol. 112, pp. 796–807, Jul. 2007.

[10] C. Jennings and J. Terpenny, “Taxonomy of Factors for Lifetime Buy,” *Ind. Syst. Eng. Res. Conf.*

[11] V. Prabhakar and P. Sandborn, “A Part Total Cost of Ownership Model for Long Life Cycle Electronic Systems,” *Int. J. Comput. Integr. Manuf.*, Jul. 2010.

[12] D. Feng, P. Singh, and P. Sandborn, “Lifetime Buy Optimization to Minimize Lifecycle Cost,” presented at the Proceedings of the 2007 Aging Aircraft Conference.

[13] P. Sandborn and R. Jafreen, “Cost Model for Assessing the Transition to Lead-Free Electronics,” *Proc. 2007 Aging Aircr. Conf.*

[14] L. Zheng and J. Terpenny, “A Hybrid Ontology Approach for Integration of Obsolescence Information,” *Comput. Ind. Eng.*, vol. 65, no. 3, pp. 485–499, Jul. 2013.

[15] X. Chang, R. Rai, and J. Terpenny, “Development and Utilization of Ontologies in Design for Manufacturing (DFM),” *J. Mech. Des.*, vol. 132, no. 2, p. 12.

[16] L. Zheng, R. Nelson, J. Terpenny, and P. Sandborn, “Ontology-Based Knowledge Representation for Obsolescence Forecasting,” *J. Comput. Inf. Sci. Eng.*, vol. 13, no. 1, 2012.

[17] R. C. Stogdill, “Dealing with Obsolete Parts,” *Des. Test Comput. IEEE*, vol. 16, no. 2, pp. 17–25.

[18] P. Sandborn, V. Prabhakar, and O. Ahmad, “Forecasting electronic part procurement lifetimes to enable the management of DMSMS obsolescence,” *51*, pp. 392–399, 2011.

[19] E. Payne, “DoD DMSMS Conference,” Charlotte, N.C., 2006.

[20] C. Josias, “Hedging Future Uncertainty: A Framework for Obsolescence Prediction, Proactive Mitigation and Management,” University of Massachusetts - Amherst, ScholarWorks, 2009.

[21] C. Josias and J. Terpenny, “Component Obsolescence Risk Assessment,” presented at the Industrial Engineering Research Conference, 2004.
[22] W. van Jaarsveld and R. Dekker, “Estimating Obsolescence Risk From Demand Data - A Case Study,” *Int. J. Prod. Econ.*, vol. 133, pp. 423–431, 2010.
[23] F. J. R. Rojo, R. Roy, and S. Kelly, “Obsolescence Risk Assessment Process Best Practice,” presented at the Journal of Physics Conference, 2012, p. 365.
[24] P. Sandborn, “A Data Mining Based Approach to Electronic Part Obsolescence Forecasting,” *IEEE Trans Compon. Packag. Technol.*, vol. 30, no. 3, pp. 397–401, 2007.
[25] P. Sandborn, F. Mauro, and R. Knox, “A Data Mining Based Approach to Electronic Part Obsolescence Forecasting,” presented at the DMSMS Conference, 2005.
[26] R. Solomon, P. Sandborn, and M. Pecht, “Electronic Part Life Cycle Concepts and Obsolescence Forecasting,” *IEEE Trans Compon. Packag. Technol.*, vol. 23, no. 1, pp. 190–193, Mar. 2000.
[27] J. Terpenny, “MIE 754: Manufacturing & Engineering Economics,” Marston Hall at UMass, 1998.
[28] D. Rink and J. Swan, “Product Life Cycle Research: A literature Review,” *J Bus Res*, vol. 14, pp. 219–242, 1979.
[29] P. Spicer, Y. Koren, M. Shpitalni, and D. Yip-Hoi, “Design Principles for Machining System Configurations,” *CIRP Ann. - Manuf. Technol.*, vol. 51, no. 1, pp. 275–280, 2002.
[30] T. M. Mitchell, “The Discipline of Machine Learning,” 2006.
[31] J. Bennett and S. Lanning, “The Netflix Prize,” 2009.
[32] J. A. Cruz and D. S. Wishart, “Applications of Machine Learning in Cancer Prediction and Prognosis,” presented at the Cancer informatics, 2007.
[33] J. Wright, “Sparse Representation for Computer Vision and Pattern Recognition,” presented at the Proceedings of the IEEE, vol. 98, pp. 1031 – 1044.
[34] C. Tucker and K. M. Harrison, “PREDICTING EMERGING PRODUCT DESIGN TREND BY MINING PUBLICLY AVAILABLE CUSTOMER REVIEW DATA,” *Int. Conf. Eng. Des.*, Aug. 2011.
[35] S. Tuarob and C. Tucker, “FAD OR HERE TO STAY: PREDICTING PRODUCT MARKET ADOPTION AND LONGEVITY USING LARGE SCALE, SOCIAL MEDIA DATA,” *ASME 2013 Int. Des. Eng. Tech. Conf. Comput. Inf. Eng. Conf.*, Aug. 2013.
[36] F. Wolinski, F. Vichot, and M. Stricker, “Using Learning-based Filters to Detect Rule-based Filtering Obsolescence,” presented at the Recherche d’Information Assistée par Ordinateur, RIAO 2000, 2000.
[37] Krzysztof Cois, P. Witold, R. Swiniarski, and L. Kurgan, *Data Mining: A Knowledge Discovery Approach*. .
[38] V. Vapnik and A. Chervonenkis, “Support Vector Network,” 1963.
[39] L. Breiman, “Random forest,” *Mach. Learn.*, vol. 45.1, pp. 5–32.
[40] W. McCulloch and W. Pitts, “A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY,” 1943.
[41] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. McLachlan, A. Ng, B. Liu, P. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, *Top 10 algorithms in data mining*. Springer Link, 2007.

[42] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.*, 2nd ed. 2009.
[43] M. Kuhn, “Package ‘caret,’” 17-Jul-2015.
[44] D. Meyer, “Package ‘e1071,’” 05-Aug-2015.
[45] A. Liaw, “Package ‘randomForest,’” 20-Feb-2015.