A Multi-Class Classification Model for Technology Evaluation

Juhyun Lee 1, Jiho Kang 2, Sangsung Park 3, Dongsik Jang 1 and Junseok Lee 4,*

1 Department of Industrial Management Engineering, Korea University, Seoul 02841, Korea; leeju@korea.ac.kr (J.L.); jang@korea.ac.kr (D.J.)
2 Machine Learning Big Data Institute, Korea University, Seoul 02841, Korea; kangmae@korea.ac.kr
3 Department of Big Data and Statistics, Cheongju University, Chungbuk 28503, Korea; hanyul@cju.ac.kr
4 MICUBE Solution, Seoul 06719, Korea
* Correspondence: js.lee@micube.co.kr; Tel.: +82-2-553-2171

Received: 29 June 2020; Accepted: 29 July 2020; Published: 30 July 2020

Abstract: This paper proposes a multi-class classification model for technology evaluation (TE) using patent documents. TE is defined as converting technology quality to its present value; it supports efficient research and development using intellectual property rights–research & development (IP–R&D) and decision-making by companies. Through IP–R&D, companies create their patent portfolios and develop technology management strategies. They protect core patents and use those patents to cooperate with other companies. In modern society, as conversion technology has been rapidly developed, previous TE methods became difficult to apply to technology. This is because they relied on expert-based qualitative methods. Qualitative results are difficult to use to guarantee objectivity. Many previous studies have proposed models for evaluating technology based on patent data to address these limitations. However, those models can lose contextual information during the preprocessing of bibliographic information and require a lexical analyzer suitable for processing terminology in patents. This study uses a lexical analyzer produced using a deep learning structure to overcome this limitation. Furthermore, the proposed method uses quantitative information and bibliographic information of patents as explanatory variables and classifies the technology into multiple classes. The multi-class classification is conducted by sequentially evaluating the value of a technology. This method returns multiple classes in order, enabling class comparison. Moreover, it is model-agnostic, enabling diverse algorithms to be used. We conducted experiments using actual patent data to examine the practical applicability of the proposed methodology. Based on the experiment results, the proposed method was able to classify actual patents into an ordered multi-class. In addition, it was possible to guarantee the objectivity of the results. This is because our model used the information in the patent specification. Furthermore, the model using both quantitative and bibliographic information exhibited higher classification performance than the model using only quantitative information. Therefore, the proposed model can contribute to the sustainable growth of companies by classifying the value of technology into more detailed categories.

Keywords: multi-class classification model; patent big data; technology evaluation; ensemble method; Bayesian optimization

1. Introduction

Intellectual property rights–research & development (IP–R&D) refers to research and development using intellectual property rights. Patents, which are a subset of intellectual property rights, are a system that legally protects the rights of inventors as compensation for sharing their technologies. Today, the role of patent-based IP–R&D is essential to the initial and final stages of technology development.
In the initial stages, companies search for core patents. After analyzing the patents, companies judge whether their technology infringes on the rights of the core patent. In the final stages of technology development, companies evaluate the value of technology in the domain. Accordingly, companies complement their patent portfolio and develop technology management strategies. Mergers and acquisitions (M&A) and lawsuits on rights infringement based on their patents are among the typical technology management strategies. The technology evaluation (TE) in the process described above supports efficient IP–R&D. TE assists in the rapid discovery of core patents in the initial stages of technology development. Moreover, it helps companies manage their portfolios by predicting the excellence of the technology developed in the final stages of technology development.

Strasser (1970) stated that TE is a systematic planning and forecasting process that delineates costs [1]. Furthermore, TE converts the quality of technology into the present value. The TE method has been approached in terms of income, market, and cost [2]. The income approach is a method of assessing the future value arising from technology as the sum of present values. It estimates the cash flow generated using the technology. The market approach is a method of evaluating the value formed between parties with the intent to transact using market information. The cost approach is a method that uses the cost of various infrastructures used to form technology. However, in modern society, technology is developed by combining various industries. For instance, a flexible display, the existing electrical-electronics industry, and the chemical materials industry work together. These technologies are evaluated by multiple complex factors [3,4]. Noh et al. (2018) proposed a framework for evaluating technology to reflect these characteristics. They argued that TE should be conducted in terms of causality, ontology, and concreteness. In addition, they pointed out the problem that the TE model may not match the results at the time of implementation. This study used variables such as potential market size. However, the variables used in the model may have different values depending on the measurement method. Therefore, the method needs to be improved by using objective documents such as patents and reflecting the information contained therein. Lee et al. (2018) argued not only that the value of technology is crucial to the licensor and licensee, but that market, technology, financial, and environment factors are crucial as well. Hence, technology needs to be evaluated from an evolutionary perspective rather than a static one.

A TE method based on patent data has been recently studied to address these issues [5–7]. Agrawal and Henderson (2002) and Shane (2002) predicted the value of technology through a survey reflecting patent information. They measured the amount of knowledge included in patents using a survey and interviewed their inventors to conduct regression analyses. However, the results vary depending on the question method and are biased towards the opinion of the inventor. Many studies using machine learning methods have been conducted to address these limitations [8–11]. In these methods, quantitative indicators, such as the number of claims and the number of cited patents, are used as explanatory variables. Patents are text data describing the technology in abstract and detailed claims. Because the scope of the patent is determined by the claims, the inventors describe the technology in detail there. Therefore, the improved TE method must reflect this characteristic.

As computing performance has improved, text mining methods that actively use language features have been studied recently. Furthermore, the method has been applied frequently in patent analysis [12–16]. Trappey et al. (2019) proposed a multi-layer neural network model that evaluates the value of technology using quantitative information, such as the number of claims and the number of citations. Because they used only the number of texts in the bibliographic information of the patent, the model could not reflect the features of the language. The word-based method is simple and intuitive. Kim et al. (2016), Uhm et al. (2017), and Kim et al. (2018) transformed words in patent documents into word-based matrices and used them as explanatory variables. However, with this method, the space of the variable is sparse and ignores the context information of the document. Kang et al. (2015) attempted to solve this problem by measuring cosine similarity in word-based matrices.

A more advanced method is to use text information at the document level using topic modeling. Patent analysis using topic modeling is a method that considers sub-technology [17–19]. Lee et al. (2018)
converted the probability of a topic into a categorical variable and used it to predict technology transfer. Therefore, it is necessary to use a natural language processing method that converts to a continuous variable. Distributed representation (DR) is a method that can compensate for the shortcomings of word and topic-based methods. DR is an embedding method designed to preserve language characteristics and contextual information by focusing on the relationships between words \[20–23\]. DR embeds an object such as a word or document into real number space. The space is designed while considering the relationship between objects. At the word level, the probability of choosing a certain word to embed considers the conditional probability of the words to the left and right. For document embedding, like word embedding, the document ID is used as a single word. Document embedding with this structure is referred to as a paragraph vector with distributed memory (PV-DM) algorithm. Topic modeling has the assumption that various topics exist in a document according to a specific probability distribution. Topic modeling repeatedly calculates the probability that documents are included in a specific topic and the probability that words are involved in a specific topic. However, DR has no assumptions about probability distribution. It simply learns a word-to-word relationship as a single neural network structure. Since it returns continuous variables compared to topic modeling, it can be used for various machine leaning algorithms. This is because topic modeling compresses information into only one category variable, but DR does not. DR is highly useful because it can infer topics like topic modeling.

We propose a TE model based on data to support efficient IP–R&D and decision-making in companies. Previous studies had a disadvantage in that the value of technology could be biased due to the subjective opinions of experts and inventors. Many models using quantitative and bibliographic information of patents have been proposed to compensate for these shortcomings. However, in the process of formalizing the bibliographic information, there were limitations that could not accurately reflect the characteristics of the language. Although patents were technology documents, they used text preprocessing that was inappropriate for processing terminology. This study proposes a data-based TE model by addressing the limitations of previous studies. It uses the quantitative and bibliographic information of a patent as variables in the predictive model for objective TE.

Moreover, bibliographic information uses a deep learning structure preprocessing method to process terminology. It uses natural language processing algorithms together to preserve contextual information. The proposed method is designed to classify the cost of technology into multiple classes using both types of patent information as explanatory variables. The cost of the technology classified as multi-class supports more detailed decision-making and efficient portfolio management of a company compared to using two classes. We introduce the concept of sequential evaluation (SE) and classify it in order of multiple classes. The concept is model-agnostic, enabling the use of diverse algorithms. The technologies evaluated using the proposed method can be compared relative to their effectiveness and thus be used in technology management strategies.

The main research questions are:

1. What is the ideal way to use a model for sustainable technology evaluation?
2. Which explanatory variables improve the prediction performance of the model?

To answer these two questions, we propose a new multi-class classification structure and use bibliographic information as an explanatory variable. The order is as follows. In Section 2, the literature background of SE is explained. Section 3 describes the proposed methodology for evaluating the technology in the order of implementation. In Section 4, we conduct an experiment to verify the applicability of the proposed methodology and derive the results. Section 5 discusses not only the strengths of the proposed method but also its shortcomings. Finally, Section 6 suggests future research directions to complement the topics discussed in the previous section.
2. Background

2.1. Literature Review of Multi-Class Classification Methods

We propose a concept for classifying patents into multiple classes. Multi-class classification models exist in various forms. The most common is One-Versus-One (OVO) or One-Versus-All (OVA) classification using a bi-class model [24–26]. OVO performs \( \frac{k(k-1)}{2} \) bi-class classification on \( k \) categories and classifies them into the most allocated category. OVA divides \( k \) categories into one category and another category, and classifies the result as a bi-class. These methods, however, have high computational costs.

Previous studies have tried to address this shortcoming by classifying data into multiple classes by linking the models in a top-down tree-based structure [27–35]. Kumar et al. (2002) and Rajan and Ghosh (2004) proposed the concept of a Binary Hierarchical Classifier (BHC) to solve multi-class classification problems using a bi-class classification model. BHC is a method of classifying \( k-1 \) models into a top-down tree structure to classify data into \( k \) categories. Ma et al. (2003) stated the center distance of each class to consider the hierarchical order in the tree structure model. They presented an algorithm to classify new data into classes based on proximity to the center of the class.

Cheong et al. (2004) proposed a multi-class classification model with a tree structure using a support vector machine (SVM). They had a structure similar to the tree structure algorithm, but it was not a model-agnostic method. Vural and Dy (2004) proposed a multi-class classification model by combining various models with a tree structure. In contrast to previous studies, they conducted clustering for each leaf of a tree and assigned the average of the clusters as a class. Although this method is model-agnostic, variability is high due to clustering. Tang et al. (2005) used the probability distribution of classes to consider the order in a multi-class hierarchical structure. They suggested an algorithm that preferentially classifies the assigned class in the tree structure based on the probability of belonging to a specific distribution. Cheng et al. (2008) presented a tree-based hierarchical structure for multi-class classification with SVM. They proposed order-based rules to use a larger classification space in the upper layer. After measuring the class similarity, the study classified it as multi-class by repeatedly using the method of creating bi-classes among similarities. Farid et al. (2014) exhibited a hybrid algorithm using decision tree and naïve Bayesian models. They proposed a method of calculating the importance of variables in the data with a decision tree and applying them to the naïve Bayesian model using only the important variables. This method was effective in reducing the computational complexity of the naïve Bayesian model. Although the previous studies can classify multiple classes with a simple structure, there is a disadvantage because it is difficult to consider the order of each class. Akoka and Comyn-Wattiau (2017) proposed a method of classifying technologies into hierarchical structures from a social and technical point of view. The structure evaluates the technology by branching features into a tree structure according to two points of view. However, the method is characterized by reflecting expert opinions to add technologies and perspectives that are not in the existing structure. Because of that, this study includes disadvantages that may bias their opinions.

Other studies were conducted using a model-agnostic approach based on probability theory [36–38]. Zohar and Roth (2001) presented a method of using the probability that data belong to each class. They assign to the class when the probability is greater than a predetermined threshold. Furthermore, sequential classification was performed by assigning an order based on the case where the reference value of the class is large. Peled et al. (2002) used a method of finding the probability of data belonging to a specific class using a logistic regression model. This study assigned to a closer class by estimating the Kullback-Leibler divergence for the probability distribution of data. Krawczyk et al. (2018) proposed a dynamic ensemble selection (DES) algorithm. DES assigns to the class of the nearest neighbor of the data, with a new voting rule in which data are not classified as only one class. This method can classify into multiple classes without being limited to a specific model.
As described in the previous studies, multi-class classification studies have been conducted using three approaches. The first approach is to repeatedly classify multi-class using the OVO or OVA method. The second is to classify and combine multiple models into tree structures. The third is to model the probability of belonging to a class. We propose a classification method for evaluating technology using multiple classes. The proposed methodology combines several models into specific structures to address the shortcomings of previous studies. Moreover, SE does not utilize a specific algorithm, but rather a structure for connecting algorithms. Thus, it is model-agnostic. In addition, the novel method is characterized by having an order of categories. Then, the value of the technology is classified into an ordered multi-class. Ordered categories have the advantage of comparing their superiority. Therefore, using this method, TE will help companies and universities to provide a detailed portfolio of patents.

2.2. Ensemble Method

Most models used in machine learning are represented by \( y = f(x) + \varepsilon \). According to the normal distribution where the error \( \varepsilon \) of the mean is zero and the variance is \( \sigma^2 \), the error of the model can be expressed by Equation (1) for each individual value.

\[
Err(x_0) = E[(Y - f(x)|x = x_0)^2],
\]

Equation (2) is equivalent to Equation (1) when developed using the features of \( y \) and the error term. In Equation (2), the first term is the square of the bias, and the second term is the variance. \( \sigma^2 \) is the irreducible variance. In machine learning, there is a trade-off between bias and variance. Models with a low bias and high variance, such as the SVM, have large variations in performance as the data change, but the values are accurate when well-tuned. Conversely, a model with high bias and low variance, such as logistic regression, has a constant performance based on the data, but the difference between the mean of the estimate and the actual value can be significant.

\[
Err(x_0) = \left( f(x_0) - \bar{f}(x_0) \right)^2 + E[\bar{f}(x_0) - \hat{f}(x_0)]^2 + \sigma^2,
\]

The ensemble method improves prediction performance by combining many weak learners [39]. It includes bagging and boosting methods. Bagging is a method of lowering the variance when combining many weak learners. Bootstrap is a technique that extracts N subsamples by sampling with replacements from a sample of size N. Bagging is an ensemble technique that uses weak learners while repeating the bootstrap technique. As shown in the left of Figure 1, bagging creates several models with N bootstrap data and classifies them according to the majority rule. Because it can use parallel computing, fast learning speed is an advantage. Its representative algorithm is a random forest (RF) [40], combined with a simple decision tree model.

![Conceptual diagram of bagging and boosting: (a) Bagging, (b) Boosting.](image)

Boosting is a method of lowering bias when combining many weak learners. As shown in the right of Figure 1, it learns many weak learners sequentially and learns to compensate by passing the error of
The ensemble method combines weak learners to create a model with excellent performance. It uses bootstrap sampling methods or combines them sequentially to use weak learners. Due to this structure, the method requires many hyperparameters, such as the number of weak learners, to be combined. The hyperparameter must be optimized to use the ensemble model. Accordingly, this study uses the Bayesian optimization method. Bayesian optimization assumes an unknown objective function that receives the input value \( x \) and finds the optimal solution \( x' \) that maximizes it \([45,46]\).

Bayesian optimization is divided into elements that estimate the function \( f \) from viewpoint \( t \) to \( x_t \) and \( x_{t+1} \). The first element is the surrogate model. The surrogate model estimates \( f \) based on \( S_t = \{(x_h, f(x_h)) | h = 1, 2, \ldots, t\} \) irradiated until viewpoint \( t \) is obtained. Bayesian optimization mainly uses the Gaussian process (GP) as a surrogate model \([47]\). GP is a probability model expressed as a probability distribution based on a combined distribution of unknown functions. GP has mean function \( \mu \) and covariance function \( k(x, x') \) as parameters. The second element is the acquisition function. This recommends \( x_{t+1} \) suitable for finding the optimal solution \( x' \) based on the results of the surrogate model. The Bayesian optimization estimates \( f \) by adding the recommended input \( x_{t+1} \) to \( S_t \) and again applying \( S_{t+1} \) to the surrogate model. Figure 2 illustrates an example of estimating a function using Bayesian optimization: the process of estimating the actual optimum searching up to \( S_{10} \). The numbers on the dots in the graph illustrate the order searched. The GP covariance values of Points 1, 3, 5, and 9 are values near the actual optimum of the graph, are small; the covariance values of points corresponding to the remaining numbers are large. Accordingly, Bayesian optimization also searches for a range that is not near the optimum value and reduces the risk of falling into the local optimum.

![Figure 2](image-url)

Figure 2. Example of the process of finding the optimum using Bayesian optimization.

In the example, it is necessary to select \( S_1 \) and then \( S_2 \). This standard is largely divided into exploitation and exploration strategies. The exploitation strategy determines \( x_{t+1} \) near the input, where the mean of \( f \) is the maximum at viewpoint \( t \). Conversely, the exploration strategy reduces risk by determining \( x_{t+1} \), where the estimated variance of \( f \) is large. They are in a trade-off relationship, and the choice of acquisition function determines what is important.

Expected improvement (EI) is an acquisition function designed to consider both strategies. EI selects \( x_{t+1} \) considering the probability that \( f^{\text{max}+} \), which is greater than the maximum value of \( f^{\text{max}} \) among \( S_t \), is derived, and the difference between the function value and \( f^{\text{max}+} \) at that time.
An improvement at \( x \), \( I(x) \), is the difference between the function value and \( f^{\text{max+}} \) for the input value \( x \) (Equation (3)).

\[
I(x) = \max(f^{\text{max+}} - f(x), 0).
\] (3)

Then, using GP, \( x \) follows a normal distribution with mean \( \mu \) and variance \( \sigma^2 \), and noise \( \epsilon \) follows standard normal distribution. \( EI(x) \) in Equation (4) is the expected value of \( I(x) \).

\[
EI(x) = E_{\epsilon \sim N(0,1)}[I(x)],
\] (4)

Because \( I(x) \) always satisfies \( f^{\text{max+}} \geq f(x) \), the integral calculation of \( EI(x) \) is as depicted in Equation (5).

\[
EI(x) = \int_{-\infty}^{(f^{\text{max+}} - \mu)} I(x) \phi(\epsilon) d\epsilon,
\] (5)

Equation (6) is the result of developing Equation (5) and adding the parameter \( \xi \) to control the relative strength of exploration and exploitation. \( \xi \) is a real number greater than zero. A greater value increases the intensity of exploration, whereas a lower value increases that of exploitation.

\[
EI(x) = \left( f^{\text{max+}} - \mu + \xi \right) \Phi \left( \frac{f^{\text{max+}} - \mu + \xi}{\sigma} \right) + \sigma \phi \left( \frac{f^{\text{max+}} - \mu + \xi}{\sigma} \right),
\] (6)

The EI method of Equation (6) finds the next input value based on whether it is possible to obtain a function value larger than the existing input values and its magnitude. Then, Bayesian optimization considers exploitation and exploration strategies. Snoek et al. (2012) presented practical Bayesian optimization guidelines demonstrating state-of-art performance using the GP surrogate model and EI function as the acquisition function using the zero-vector as the mean function and the Matérn 5/2 kernel as the covariance function.

3. Proposed Methodology

This paper proposes a model for evaluating the cost of patents. The flowchart in Figure 3 illustrates the proposed methodology. First, the analyst collects patent documents of the target technology domain. They extract quantitative and textual information from the patents.

![Figure 3. Flowchart of proposed methodology.](image)

The lexical analysis and DR algorithms convert the extracted bibliographic information. The lexical analysis divides text by parts of speech. Parts of the text that are not needed are discarded. The DR algorithm projects it into the \( d \) dimensional real number space. The bibliographic information is then...
converted into a d dimensional explanatory variable. Next, the proposed model combines the variables with quantitative information. The model uses it as explanatory variables. This ensures that the model produces objective results. As an illustration, the value is the cost determined at the time of technology transfer or decided by the company. Thereafter, the splitting point generates a bi-class classification model. If the data are greater than the splitting point, the first category is assigned; otherwise, the other category is assigned. The model learns to evaluate patents in two classes. For example, suppose the splitting point is the median of cost. Then, the costs less than the median are allocated to one category, and vice versa. The model performs the task of classifying the two categories. Hyperparameters of the model are optimized using a Bayesian method. Finally, SE connects bi-class classification models into multi-class classification models. SE is a method of resampling the data evaluated to be smaller than the $i^{th}$ splitting point, reclassifying it as the $(i+1)^{th}$ point. This process is repeated until $i$ becomes $k–1$. Then, the data of unknown cost can be evaluated with $k$ categories. For example, suppose the splitting points are the third quartile and the first quartile. Then the third quartile is the 1st splitting point because it is larger than the first quartile. Data evaluated by the model as being smaller than the 1st splitting point are compared with the 2nd splitting point. Through this process, data are classified into three categories. The categories can be sorted according to the value of the splitting point. Thus, the outputs are ordered categories. This allows the value of technology to be compared to each other.

Table 1 describes the symbols used in this chapter. $d$ is a symbol used when preprocessing bibliographic information. The remaining symbols are related to SE. The preprocessing method of bibliographic information used in the above process is described in detail in Section 3.1. Model optimization and SE are described in Sections 3.2 and 3.3.

| Symbols | Description |
|---------|-------------|
| $d$     | Number of dimensions of real number space through DR |
| $P^{(i)}$ | Point at a cost equal to the first quartile ($P^{(i)} > P^{(i+1)}$) |
| $C_{P^{(i)}, C_{P^{(i)}}}$ | Cost is classified as $C_{P^{(i)}}$ if greater than $P^{(i)}$ and $C_{P^{(i)}}$ if less |
| $M_{P^{(i)}}$ | Model classified as $C_{P^{(i)}}$ if greater than $P^{(i)}$ and $C_{P^{(i)}}$ otherwise |

3.1. Preprocessing Text Information

The bibliographic information of the patent is applied to the proposed method after lexical analysis, which converts text information into the form required by machine learning algorithms. The lexical analysis proceeds in the order of tokenization, morphological analysis, and part-of-speech (POS) tagging. In English, it is often tokenized through blank lines. In contrast to English, agglutinative languages in which postposition particle and end of a word have developed are tokenized with blank lines, and morphological analysis is not properly performed [48,49].

Korean is one of the representative agglutinative languages. Many dictionary-based morpheme analyzers have been developed for agglutinative languages such as Korean [50]. However, a patent document often contains a great deal of jargon. Dictionary-based morpheme analyzers have difficulty in accurately managing jargon that is not included in the rules. Recently, a data-based morpheme analyzer with a deep learning structure was developed to compensate for this drawback. Kakao Hangul Analyzer III (Khaiii) is a representative data-based morpheme analyzer [51]. Khaiii is known to be more efficient for patent analysis than other morphological analyzers [52].

Figure 4 illustrates the conceptual diagram of an agglutinative language processor that combines a data-based morphological analyzer and PV-DM algorithm used in this study. Based on the conceptual diagram, it is possible to efficiently embed documents by performing DR to the $d$ dimension, except for any unnecessary POS information.
This is a bi-class classification model that determines whether the cost is greater or lower than the $P_{ij}$.

### 3.2. Optimizing the Models for Each Splitting Point

The proposed method classifies the cost of each technology into one of $k$ grades. The $k-1$ splitting points are determined from the cost of the patent. Each model is trained with $k-1$ splitting points. This is a bi-class classification model that determines whether the cost is greater or lower than the $i^{th}$ point.

Figure 5 illustrates the process of optimizing the bi-class classification model for each splitting point of cost. First, the data are divided into training and testing data. $P^{(i)}$ is one point of the cost of training data. For example, $P^{(i)}$ can be the first quartile of the cost. In $P^{(i)}$, $i$ is the order of the splitting points. The splitting point is the first quartile, the median, and the third quartile; for instance, $P^{(1)}$ is the third quartile and $P^{(3)}$ is the first quartile. The bi-class is the result of comparing the cost of training data with the splitting point. $M_{p(i)}$ is the model that classifies data into a bi-class based on $P^{(i)}$. $C_{p^{(i)}}$ is the category assigned when the cost of the data is predicted by $M_{p(i)}$ to be greater than $P^{(i)}$. $C_{p^{(i)}}$ is the category assigned when the cost of the data is predicted by $M_{p(i)}$ to be lower than $P^{(i)}$. The bi-class classification model optimizes the hyperparameter through comparison with testing data. According to the proposed method, the number of hyperparameters increases with the number of models and the number of classes, $k$. If there are five models and $k$ is three, simple fitting requires 15 iterations. Thus, the proposed method uses Bayesian-optimization to solve the problem of increasing the number of hyperparameters.

![Conceptual diagram of bi-class classification models when $k = 4$.](image)

### 3.3. Sequentially Evaluating the Technology

The proposed method optimizes the model for each splitting point and then evaluates the value using the $k-1$ optimized models. SE proceeds as depicted in Figure 6.

![Conceptual diagram of the agglutinative language processor.](image)
3.2. Optimizing the Models for Each Splitting Point

The proposed method classifies the cost of each technology into one of k grades. The k–1 splitting points are determined from the cost of the patent. Each model is trained with k–1 splitting points. This is a bi-class classification model that determines whether the cost is greater or lower than the 𝑖th point.

Figure 5 illustrates the process of optimizing the bi-class classification model for each splitting point of cost. First, the data are divided into training and testing data. 𝑃(𝑖) is one point of the cost of training data. For example, 𝑃(𝑖) can be the first quartile of the cost. In 𝑃(𝑖), i is the order of the splitting points. The splitting point is the first quartile, the median, and the third quartile; for instance, 𝑃(3) is the third quartile and 𝑃(1) is the first quartile. The bi-class is the result of comparing the cost of training data with the splitting point. 𝑀 mould(𝑖) is the model that classifies data into a bi-class based on 𝑃(𝑖). 𝐶 mould(𝑖) is the category assigned when the cost of the data is predicted by 𝑀 mould(𝑖) to be greater than 𝑃(𝑖). 𝐶 mould(𝑖) is the category assigned when the cost of the data is predicted by 𝑀 mould(𝑖) to be lower than 𝑃(𝑖). The bi-class classification model optimizes the hyperparameter through comparison with testing data. According to the proposed method, the number of hyperparameters increases with the number of models and the number of classes, k. If there are five models and k is three, simple fitting requires 15 iterations. Thus, the proposed method uses Bayesian-optimization to solve the problem of increasing the number of hyperparameters.

![Figure 5. Conceptual diagram of bi-class classification models when k = 4.](image)

This example is the process of classifying into four grades. The largest splitting point is 𝑃(1), which satisfies Equation (7).

\[ p(i) > p(i+1), \quad p(k) = 0, \quad i = 1, 2, \ldots, k-1, \quad (7) \]

Next, SE evaluates the cost with 𝑀 mould(1) and resamples only the data allocated in the 𝐶 mould(1) category. The data are again evaluated with 𝑀 mould(2). SE repeats the above process until i becomes k–1. Accordingly, the technology cost is classified into k grades.

4. Experiment Result

4.1. Data Description

This chapter describes the experiments with the proposed method using actual data. The 232 patents used in the experiment are actual technology transfer data from University A. A list of variables used in the experiment is depicted in Table 2.

![Table 2. Variables used in the proposed model.](image)

| Variables | Description |
|-----------|-------------|
| Citation  | The number of forward citations |
| Claim     | The number of registered claims |
| Family    | The number of family countries |
| IPC       | The number of IPC codes |
| Registration | Registration status (dummy) |
| Uncertainty | The number of independent claims/(The number of cited patents+1) |
| 𝐷𝑅1, \ldots, 𝐷𝑅𝑑 | \(d\) variables obtained because of distributed representation |
| Transfer cost | The cost of technology transfer |

In our experiment, the quantitative information mean the explanatory variables, such as citation, claim, family, IPC, registration, and uncertainty [53]. DR is a real number in \(d\) dimensional semantic space derived through document embedding. The technology transfer cost was used as the value of the patent. The cost is set by the applicant considering the level of technology. Purchaser trade by considering the value and cost of technology. Therefore, we use the cost of technology transfer as the value of technology [54,55].
4.2. Experimental Study

At this stage of the experiment, we collected the patent documents of University A. Because the patent document is a technology document with lots of jargon, it is necessary to process words that are not included in the dictionary more accurately. Accordingly, the experiment used an agglutinative language processor (in Figure 4). Furthermore, the patent document was embedded in an eight-dimensional real number space with the processor.

Models for evaluating technology costs were designated as ensemble-based AB, GB, and XGB. Three splitting points of the technology transfer cost of the collected data were designated to evaluate the technology value with the ensemble model. Three splitting points are depicted in Table 3: the number of third quartiles, median, and the number of first quartiles of the transfer cost. The performance measures are accuracy, precision, and specificity. Accuracy is the fitting ratio of all bi-classes. Precision and specificity are the ratios of actual $C_{p(i)}$ and $C_{p\neg(i)}$ predicted by the model, respectively. SE does not re-evaluate data classified as $C_{p(i)}$; and only extracts the data classified as $C_{p\neg(i)}$. Because of this structure, in $C_{p(i)}$, precision should be used to monitor whether the model accurately classifies the data into it. For $C_{p\neg(i)}$, it is necessary to monitor whether the actual $C_{p\neg(i)}$ is accurately classified based on specificity.

| Splitting Point | Description |
|-----------------|-------------|
| $p^{(1)}$       | $Q_3$, Number of third quartiles of transfer cost |
| $p^{(2)}$       | $Q_2$, Number of second quartiles of transfer cost |
| $p^{(3)}$       | $Q_1$, Number of first quartiles of transfer cost |

Three models were optimized with a Bayesian approach for three splitting points. For Bayesian optimization, the surrogate model used a GP with a mean function of zero-vector, a covariance function of Matérn 5/2 kernel, and an acquisition function of EI with a $\xi$ of 0.01. This Bayesian optimization sets the accuracy measure of the model as an objective function. Figure 7 visually presents the prediction interval of the performance measure according to the splitting points and models. The prediction intervals were determined based on the assumption that the degree of freedom is 9, and the significant level is 0.05, which follows the $T$-distribution. For the model optimized using the Bayesian method, the average and standard error of each performance measure were obtained using 10-cross validation.

![Figure 7. Comparison of prediction interval according to splitting points and models.](image)

Based on the experiment results, $Q_3$ the first splitting point, XGB had the highest scores for all performance measures. When $Q_2$ was the splitting point, the accuracy was the highest in GB, but the rest of the performance measures were excellent in XGB. When $Q_1$ is the splitting point, all performance measures of XGB had the highest scores. SE extracts data smaller than the first splitting point and evaluates it again as the second splitting point. Due to this structure, the precision of the model is important for SE. Therefore, XGB with the highest precision was used as the final model for all splitting points.
Next, the optimal model for the three splitting points classifies the data into four classes using the SE method. The data were divided into 7-to-3 ratios to confirm the final classification performance of the proposed model. \( Q_1 \) of training data is 30,000,000 Won, \( Q_2 \) is 12,000,000 Won, and \( Q_3 \) is 9,318,182.5 Won (Won is the currency unit of Korea). A boxplot of the cost of training data is depicted in Figure 8.

![Figure 8. Comparison of cost distribution of training data for each class.](image)

\( C_{P+1} \) is a class assigned when \( M_{p(1)} \) produces a value of technology greater than \( P^{(1)} \) (\( = Q_3 \)). \( C_{P+2} \) is a case where the technology cost is predicted to be less than or equal to \( P^{(1)} \) but greater than \( P^{(2)} \) (\( = Q_2 \)). \( C_{P+3} \) is a case where the cost is less than or equal to \( P^{(2)} \) but greater than \( P^{(3)} \) (\( = Q_1 \)). Finally, \( C_{P-3} \) is the case where the cost is less than or equal to \( P^{(3)} \). Table 4 illustrates the performance measure of each splitting point and the final multi-class classification.

| Class       | The Proposed Model | Model without Text Information |
|-------------|-------------------|-------------------------------|
|             | Accuracy | Precision | Specificity | Accuracy | Precision | Specificity |
| \( C_{P+1} \) | 0.843    | 0.000     | 1.000      | 0.843    | 0.000     | 1.000      |
| \( C_{P+2} \) | 0.329    | 0.312     | 0.064      | 0.529    | 0.333     | 0.574      |
| \( C_{P+3} \) | 0.671    | 1.000     | 1.000      | 0.500    | 0.310     | 0.565      |
| \( C_{P-3} \) | 0.786    | 0.083     | 0.831      | 0.671    | 0.000     | 0.810      |
| Macro       | 0.657    | 0.349     | 0.724      | 0.636    | 0.161     | 0.737      |
| Micro       | 0.657    | 0.286     | 0.747      | 0.636    | 0.271     | 0.757      |

The proposed method classifies technology into four classes. The final performance of the model is measured by macro-precision and micro-precision. The macro-precision is the arithmetic mean result of each class. For example, the macro-precision is the mean of precision for all classes. Micro-precision aggregates the contributions of all classes. This calculates the sum of values such as true positive and false negative for all classes. The result of the calculation is the same as the confusion matrix of the binary classification, and the micro-precision is the performance value obtained from it. The accuracy based on the proposed method is 0.657 for both macro and micro. The macro-precision and macro-specificity are 0.349 and 0.724, and the micro-precision and micro-specificity are 0.286 and 0.747, respectively. The performance of the proposed method compared the performance with a model using only quantitative information. Consequently, the proposed model exhibited higher accuracy and precision than the model using only quantitative information. As described in the previous section, the precision of the model is essential to SE. Therefore, a multi-class evaluation of technology using SE must use both quantitative and text information.

The developed model can be used for various purposes, such as the following scenario:
1. Licensing strategy: A company hopes to foray into a new business. To achieve this, they require a patent owned by another company. However, there is insufficient time and technology to develop related patents. The TE model deduces that the technology of other company is excellent. The first company obtains permission to implement the technology through licensing or M&A.

2. Initial stages of IP-R&D: The company tries IP-R&D. The problem is that they have to circumvent the core patents of the domain. The company collected patents and predicted their value through the TE model. They were able to conduct design circumvention by filtering high-value patents.

3. Final stages of IP-R&D: The company applies for a new patent through scenario 2. However, they wondered how excellent this patent would be in the domain. They predicted its value with the TE model. The patent was predicted to be of high value. Therefore, they observe the development of new patents that violate its rights.

The scenarios mentioned above are just some of the ways in which the proposed model can be used. We expect this model to be used in a variety of ways.

5. Discussion

TE is essential for efficient IP–R&D. The importance of data-based TE has been emphasized in recent years as technologies cross converging industries are developed. Data-based TE has developed in conjunction with patent analysis. Patents represent big data that is massive, rapidly generated, and diverse in form. Among this big data, core patents are used for corporate M&A, technology commercialization, technology trading, and transfer as part of IP–R&D activities. A TE model based on patent big data is required for this application.

Previous studies have used patent quantitative and text information as explanatory variables for data-based TE. It is important to convert text information. The studies were mainly conducted by using word-based and topic-based methods. However, the word-based method does not consider the context information of a sentence. Furthermore, the topic-based method is difficult to apply because text information is returned as a categorical variable. We used DR to overcome this shortcoming. DR is also used with language processors to take into account the features of natural language. Because the patent document contains a great deal of jargon, a lexical analyzer with a deep learning structure rather than a dictionary is used.

We proposed a model for evaluating the value of technology. The methodology classifies the value of patents into multiple classes. This was designed to evaluate the cost of patents by reflecting quantitative information and bibliographic information. At the stage of reflecting the bibliographic information, a processor was designed to consider the features of the language. A lexical analyzer based on deep learning was combined to increase the accuracy of the analysis and improve the characteristics of the patent document.

Each domain has distinct characteristics that are ignored by TE using only quantitative information. The proposed model uses text information and quantitative information. Furthermore, it is versatile because it classifies values into multiple classes. Companies should be able to evaluate their technology using this method.

Our research has limitations that will be described below. The first limitation is the problem of determining the splitting point. The splitting point was used to divide the value of technology into categories. In the experiment, it was determined by the first quartile, the median, and the third quartile of the cost of technology transfer. In this process, the splitting point used general-purpose statistics, but it is necessary to study how to determine the appropriate number. The second limitation is for multilinguals. In the experiment, a patent developed in Korean was used. However, TE needs to utilize patents from several countries because it needs to progress dynamically according to market formation. The proposed model can only be applied to a single language, so this should be improved in the future. The last problem is when the cost of value exists in various units. In the same context as previously pointed out, the integration of cost units in different countries should be considered. These problems need to be solved so the proposed model can be generated sufficient results when used.
6. Conclusions

In modern society, the development of convergence technology occurs rapidly. Hence, traditional TE methods have various limitations. Consequently, many data-based TE methods have been studied. Patents are data with features that have quantitative and bibliographic information, all of which are required by advanced TE methods. The proposed methodology is used as an explanatory variable by considering both types of information. It performs TE using a new multi-class classification concept. Although SE is model-agnostic because it connects multiple models sequentially, it requires higher computational complexity depending on the number of models and classes. We have combined the Bayesian optimization approach to address this shortcoming. With Bayesian optimization, SE has been improved to apply various models regardless of the number of hyperparameters in the predictive model.

This study conducted experiments using actual patents to confirm the applicability of the proposed method. In the experiment, the first quartile, median, and third quartile of the technology transfer cost were used as splitting points. The experiment used a model based on an ensemble algorithm to ensure generalized performance. The ensemble model was optimized with a Bayesian approach for three splitting points. After optimization, XGB was suitable for multi-class classification using SE. Based on the results of multi-class classification, the model, including text information, has higher accuracy and precision than the model only using quantitative information. These results confirmed that the inclusion of text information is suitable for evaluating technical value.

Future research needs to be conducted to overcome the limitations discussed in the previous section. The three points discussed are the matter of determining the splitting point, processing multilinguals, and integrating various units. The first point can be approached by estimating the probability distribution of cost. It is possible to measure the skewness, kurtosis, and central tendency of the probability distribution and to determine the optimal number and value of splitting points. The rest of the points will have to be solved by combining multidisciplinary methods. In particular, the multilingual problem can be approached with various attempts, as the natural language processing algorithm has recently been advanced. If the limitations discussed are resolved in future studies, then the proposed method is expected to be promoted to a TE model that is applicable to the global market.

Author Contributions: J.L. (Juhyun Lee), J.K., and J.L. (Junseok Lee) conceived and designed the experiments; S.P. and D.J. analyzed the data to illustrate the validity of this study; J.L. (Juhyun Lee) wrote the paper and performed all of the research steps. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Republic of Korea government (MSIT) (No. NRF-2020R1A2C1005918).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Strasser, G. Developing a Technology Assessment Capability; Office of Science and Technology, Executive Office of the President: Washington, DC, USA, 1970.
2. Korea agency for infrastructure technology advancement. Manual for technology valuation. In Ministry of Land, Infrastructure and Transport; Korea Agency for Infrastructure Technology Advancement: Seoul, Korea, 2018.
3. Noh, H.; Seo, J.; Yoo, H. How to improve a technology evaluation model: A data-driven approach. Technovation 2018, 72, 1–12. [CrossRef]
4. Lee, J.; Sung, T.; Kim, E.; Shin, K. Evaluating determinant priority of license fee in biotech industry. J. Open Innov. Technol. Mark. Complex. 2018, 4, 30. [CrossRef]
5. Banerjee, A.; Bakshi, R.; Sanyal, M.K. Valuation of patent: A classification of methodologies. Res. Bull. USA 2017, 42, 158–174.
6. Agrawal, A.; Henderson, R. Putting patents in context: Exploring knowledge transfer from MIT. Manag. Sci. 2002, 48, 44–60. [CrossRef]
7. Shane, S. Selling university technology: Patterns from MIT. Manag. Sci. 2002, 48, 122–137. [CrossRef]
8. Trappey, A.J.; Trappey, C.V.; Wu, C.Y.; Lin, C.W. A patent quality analysis for innovative technology and product development. *Prog. Adv. Comput. Intell. Eng.* 2012, 26, 26–34. [CrossRef]

9. Yang, D.; Kim, S.; Kang, G. Some Methods Determining Reasonable Royalty Rates for Patent Valuation-An Infringement Damages Model. *J. KTIS* 2012, 15, 700–721.

10. Sohn, S.; Lee, W.; Ju, Y. Valuing academic patents and intellectual properties: Different perspectives of willingness to pay and sell. *Technovation* 2013, 33, 13–24. [CrossRef]

11. Woo, H.; Kwak, J.; Lim, C. A study on patent evaluation model based on Bayesian approach of the structural equation model. *KIAS* 2017, 30, 901–916.

12. Trappey, A.J.; Trappey, C.V.; Govindarajan, U.H.; Sun, J.J. Patent value analysis using deep learning models-the case of IoT technology mining for the manufacturing industry. *IEEE Trans. Eng. Manag.* 2019, 1–13. [CrossRef]

13. Kim, J.; Jun, S. Zero-inflated poisson and negative binomial regressions for technology analysis. *Int. J. Softw. Eng. Appl.* 2016, 10, 431–448. [CrossRef]

14. Uhm, D.; Ryu, J.; Jun, S. An Interval Estimation Method of Patent Keyword Data for Sustainable Technology Forecasting. *Sustainability* 2017, 9, 2025. [CrossRef]

15. Kim, J.; Choi, J.; Park, S.; Jang, D. Patent keyword extraction for sustainable technology management. *Sustainability* 2018, 10, 1287. [CrossRef]

16. Kang, P.; Geum, Y.; Park, H.; Kim, S.; Sung, T.; Lee, H. A Market-Based Replacement Cost Approach to Technology Valuation. *J. KIE* 2015, 41, 150–161.

17. Kim, J.; Lee, J.; Kim, G.; Park, S.; Jang, D. A hybrid method of analyzing patents for sustainable technology management in humanoid robot industry. *Sustainability* 2016, 8, 474. [CrossRef]

18. Kang, J.; Lee, J.; Jang, D.; Park, S. A Methodology of Partner Selection for Sustainable Industry-University Cooperation Based on LDA Topic Model. *Sustainability* 2019, 11, 3478. [CrossRef]

19. Lee, J.; Kang, J.; Jun, S.; Lim, H.; Jang, D.; Park, S. Ensemble modeling for sustainable technology transfer. *Sustainability* 2018, 10, 2278. [CrossRef]

20. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems: Lake Tahoe, NV, USA, 2013; pp. 3111–3119.

21. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* 2013, arXiv:1301.3781.

22. Dai, A.M.; Olah, C.; Le, Q.V. Document embedding with paragraph vectors. *arXiv* 2015, arXiv:1507.07998.

23. Le, Q.; Mikolov, T. Distributed representations of sentences and phrases and their compositionality. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems: Lake Tahoe, NV, USA, 2013; pp. 3111–3119.

24. Aly, M. Survey on multi-class classification methods. *Neural Netw.* 2005, 19, 1–9.

25. Sejnowski, T.J.; Rosenberg, C.R. Parallel networks that learn to pronounce English text. *Complex Syst.* 1987, 1, 145–168.

26. Lorena, A.C.; De Carvalho, A.C.; Gama, J.M. A review on the combination of binary classifiers in multi-class problems. *Artif. Intell. Rev.* 2008, 30, 19. [CrossRef]

27. Kumar, S.; Ghosh, J.; Crawford, M.M. Hierarchical fusion of multiple classifiers for hyperspectral data analysis. *Pattern Anal. Appl.* 2002, 5, 210–220. [CrossRef]

28. Rajan, S.; Ghosh, J. An empirical comparison of hierarchical vs. two-level approaches to multi-class problems. In *International Workshop on Multiple Classifier Systems*; Springer: Berlin, Germany, 2004; pp. 283–292.

29. Ma, X.X.; Huang, X.Y.; Chai, Y. 2PTMC classification algorithm based on support vector machines and its application to fault diagnosis. *Control Decis.* 2003, 18, 272–276.

30. Cheong, S.; Oh, S.; Lee, S. Support vector machines with binary tree architecture for multi-class classification. *Neural Inf. Process. Lett. Rev.* 2004, 2, 47–51.

31. Vural, V.; Dy, J.G. A hierarchical method for multi-class support vector machines. In *Proceedings of the Twenty-First International Conference on MACHINE Learning*, Banff, AB, Canada, 4–8 July 2004.

32. Tang, F.M.; Wang, Z.D.; Chen, M.Y. On multi-class classification methods for support vector machines. *Control Decis.* 2005, 20, 746. [CrossRef]

33. Cheng, L.; Zhang, J.; Yang, J.; Ma, J. An improved hierarchical multi-class support vector machine with binary tree architecture. In *Proceedings of the IEEE 2008 International Conference on Internet Computing in Science and Engineering*, Harbin, China, 28–29 January 2008.
34. Farid, D.M.; Zhang, L.; Rahman, C.M.; Hossain, M.A.; Strachan, R. Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks. *Expert Syst. Appl.* **2014**, *41*, 1937–1946. [CrossRef]
35. Akoka, J.; Comyn-Wattiau, I. A method for emerging technology evaluation: Application to blockchain and Smart Data Discovery. In *Conceptual Modeling Perspectives*; Springer: Cham, Switzerland, 2017; pp. 247–258.
36. Even-Zohar, Y.; Roth, D. A sequential model for multi-class classification. *arXiv* **2001**, arXiv:cs/0106044.
37. Har-Peled, S.; Roth, D.; Zimak, D. Constraint Classification: A New Approach to Multi-Class Classification and Ranking. In *Proceedings of the Neural Information Processing Systems*, Vancouver, CO, Canada, 9–14 December 2002; pp. 809–816.
38. Krawczyk, B.; Galar, M.; Woźniak, M.; Bustince, H.; Herrera, F. Dynamic ensemble selection for multi-class classification with one-class classifiers. *Pattern Recognit.* **2018**, *83*, 34–51. [CrossRef]
39. Park, E. Statistical Performance of Ensemble Methods in Constructing a Prediction Model. Master’s Thesis, School of Medicine, Biostatistics, University of Korea, Seoul, Korea, 2019.
40. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32.
41. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*; Springer: Berlin, Germany, 1995; pp. 23–37.
42. Schapire, R.E.; Singer, Y. Improved boosting algorithms using confidence-rated predictions. *Mach. Learn.* **1999**, *37*, 297–336.
43. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
44. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
45. Brochu, E.; Cora, V.M.; De Freitas, N. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv* **2010**, arXiv:1012.2599.
46. Snoek, J.; Larochelle, H.; Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems*; 2012; pp. 2951–2959. Available online: http://papers.nips.cc/paper/4522-practical-bayesian-optimization (accessed on 30 July 2020).
47. MacKay, D.J. Introduction to Gaussian processes. *NATO ASI Series F Comput. Syst. Sci.* **1998**, *168*, 133–166.
48. Hong, J.; Cha, J. Error Correction of Sejong Morphological Annotation Corpora using Part-of-Speech Tagger and Frequency Information. *J. KIISE* **2013**, *40*, 417–428.
49. Shim, K. Morpheme Restoration for Syllable-based Korean POS Tagging. *J. KIISE* **2013**, *40*, 182–189.
50. Park, E.; Cho, S. KoNLPy: Korean natural language processing in Python. In Proceedings of the 26th Annual Conference on Human & Cognitive Language Technology, Chuncheon, Korea, 10–11 October 2014; pp. 133–136.
51. khaiii, Github. 2018. Available online: https://github.com/kakao/khaiii (accessed on 15 June 2020).
52. Lee, Y.; Kim, S.; Hong, H.; Gim, J. Comparison and Evaluation of Morphological Analyzer for Patent Documents. In Proceedings of the Korean Institute of Information Technology, Daejeon, Korea, 13–15 June 2019; pp. 264–265.
53. Kwon, S.; Drev, M. Defensive Patent Aggregators as Shields against Patent Assertion Entities? Theoretical and Empirical Analysis. *Technol. Forecast. Soc.* **2020**, *151*, 119745. [CrossRef]
54. Jensen, O.W.; Scheraga, C.A. Transferring technology: Costs and benefits. *Technol. Soc.* **1998**, *20*, 99–112. [CrossRef]
55. Baek, D.H.; Sul, W.; Hong, K.P.; Kim, H. A technology valuation model to support technology transfer negotiations. *R D Manag.* **2007**, *37*, 123–138. [CrossRef]