A Real Estate Valuation Model using Boosted Feature Selection

Mr. Kankawee Chanasit

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Computer Engineering
Department of Computer Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2020

โมเดลประเมินมูลค่าอสังหาริมทรัพย์ด้วยวิธีการคัดเลือกตัวแปรแบบส่งเสริม

นายกานต์กวี ชนะสิทธิ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2563

Thesis Title	A Real Estate Valuation Model using Boosted Feature Selection
By	Mr. Kankawee Chanasit
Field of Study	Computer Engineering
Thesis Advisor	Associate Professor PROADPRAN PUNYABUKKANA, Ph.D.
Thesis Co Advisor	Associate Professor ATIWONG SUCHATO, Ph.D.

Accepted by the FACULTY OF ENGINEERING, Chulalongkorn University in Partial Fulfillment of the Requirement for the Master of Engineering

..... Dean of the FACULTY OF ENGINEERING

(Professor SUPOT TEACHAVORASINSKUN, Ph.D.)

THESIS COMMITTEE

..... Chairman

(Associate Professor CHOTIRAT RATANAMAHATANA, Ph.D.)

..... Thesis Advisor

(Associate Professor PROADPRAN PUNYABUKKANA, Ph.D.)

..... Thesis Co-Advisor

(Associate Professor ATIWONG SUCHATO, Ph.D.)

..... Examiner

(EKAPOL CHUANGSUWANICH, Ph.D.)

..... External Examiner

(Pipop Thienprapasith, Ph.D.)

กานต์กวี ชนะสิทธิ์ : โมเดลประเมินมูลค่าอสังหาริมทรัพย์ด้วยวิธีการคัดเลือกตัวแปรแบบส่งเสริม. (A Real Estate Valuation Model using Boosted Feature Selection) อ.ที่ปรึกษาหลัก : รศ. ดร.โปรดปราน บุณยพุกกณะ, อ.ที่ปรึกษาร่วม : รศ. ดร.อติวงศ์ สุชาโต

งานวิจัยด้านการประเมินราคาอสังหาริมทรัพย์นั้นนิยมใช้เทคโนโลยีโครงข่ายประสาทเทียม ซึ่งส่งผลดีเมื่อข้อมูลมีตัวแปรจำนวนมากและหลายมิติ เช่นข้อมูลราคาบ้านในประเทศสหรัฐอเมริกา อย่างไรก็ดี ข้อมูลด้านอสังหาริมทรัพย์ที่มีตัวแปรจำนวนมากนั้นไม่ได้หาได้ง่าย เช่น ข้อมูลในประเทศไทยที่มีตัวแปรน้อย งานวิจัยนี้จึงมุ่งเน้นที่จะแก้ปัญหาด้วยการวิเคราะห์ความสำคัญของตัวแปรโดยนำอัลกอริทึมของการ์สัน (Garson's algorithm) มาทำงานร่วมกับกลยุทธ์แบบส่งเสริม (boosting strategy) เพื่อสร้างกระบวนการคัดเลือกตัวแปรแบบใหม่ ที่สามารถคำนวณค่าความสำคัญของตัวแปรในโครงข่ายประสาทเทียมและปรับปรุงเงื่อนไขในการคัดเลือกจากค่าความผิดพลาดของการคำนวณในครั้งก่อนได้ในทุกขั้นตอนการทำงานบนโครงข่ายประสาทเทียม กระบวนการที่นำเสนอนี้ได้ถูกนำไปทดสอบและเปรียบเทียบผลกับวิธีการคัดเลือกตัวแปรอื่น ๆ ที่เป็นที่นิยม โดยใช้ข้อมูลสังเคราะห์และข้อมูลราคาบ้านที่ใช้งานจริงจากบริษัทโฮมดอทเทค ข้อมูลราคาบ้านในบอสตันและข้อมูลจากการแข่งขันการประเมินราคาบ้านของซิลโล่ว์ (Zillow) ในแคกเกิล (Kaggle) ผลลัพธ์ที่ได้แสดงให้เห็นว่าวิธีการของงานวิจัยนี้สามารถคัดเลือกตัวแปรที่มีผลต่อราคาบ้านได้ครบถ้วน และได้ชุดของตัวแปรที่ส่งผลต่อราคาของอสังหาริมทรัพย์ที่เฉพาะเจาะจงในแต่ละพื้นที่ และยังแสดงให้เห็นการปรับปรุงประสิทธิภาพของโมเดลในกรณีที่ข้อมูลมีจำนวนเหมาะสมกับการทำงานของโครงข่ายประสาทเทียม ผลลัพธ์ของงานวิจัยนี้เมื่อทดสอบกับข้อมูลกรณีบอสตัน (Boston Housing) ให้ค่าความผิดพลาด (error rate) 3.673 ซึ่งดีกว่าวิธีการสารสนเทศรวม (mutual information) ที่มีค่าความผิดพลาด 3.745 สำหรับชุดข้อมูลฟรีดแมน ค่าความผิดพลาดของงานวิจัยนี้ได้ 0.861 ซึ่งเทียบเคียงกับวิธีการสารสนเทศรวม ผลของงานวิจัยนี้ได้จัดอยู่ในอันดับร้อยละ 24 ต้นของการแข่งขันประเมินราคาของซิลโลว์อีกด้วย

สาขาวิชา	วิศวกรรมคอมพิวเตอร์	ลายมือชื่อนิสิต .....
ปีการศึกษา	2563	ลายมือชื่อ อ.ที่ปรึกษาหลัก .....
		ลายมือชื่อ อ.ที่ปรึกษาร่วม .....

6170113521 : MAJOR COMPUTER ENGINEERING

KEYWORD: Artificial Neural Network, Boosting, Real Estate Valuation, Sensitivity Analysis

Kankawee Chanasit : A Real Estate Valuation Model using Boosted Feature Selection. Advisor: Assoc. Prof. PROADPRAN PUNYABUKKANA, Ph.D. Co-advisor: Assoc. Prof. ATIWONG SUCHATO, Ph.D.

To estimate real estate values, a complex valuation model based on artificial neural network (ANN) has been established as a successful means in modern machine learning research, specifically when high-dimensional data are available. Unfortunately, the real estate data in many locations, such as Thailand, are quite limited in terms of features. Hence, it becomes mandatory to reduce the complexity using feature selection techniques. These techniques aim to improve performance by identifying significant factors and help decrease the computational overload and model construction. However, due to the lack of explicability and interpretability in ANNs, the analysis of input factors cannot be explained directly by model composition. In this research, we apply a combination of a boosting strategy and input sensitivity analysis in an improved Garson's algorithm to perform feature selection that can adjust its selection criteria through each iteration on an ANN model. This proposed technique is then compared with other traditional feature selection techniques using synthetic data and real-world house valuation data. The results show that our model can maintain the sensitivity coefficient for every informative feature. The technique of this study provides a set of features that influences the house price and implies the character of each specific area. It is placed among the top 24% in Zillow Prize competition

Field of Study:	Computer Engineering	Student's Signature .....
Academic Year:	2020	Advisor's Signature .....
		Co-advisor's Signature .....

ACKNOWLEDGEMENTS

The authors would like to express their sincere thanks to my advisor and co-advisors whose expertise was invaluable in formulating the research questions and methodology. Their insightful feedback pushed this work to a higher improvement. This research was partially supported by Home Dot Tech Co., Ltd., and home.co.th

Kankawee Chanasit

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

1. Introduction

Real estate industry involves some of the most massive investments in the world due to its value and economic impact. In recent years, the real estate industry has become more computerized with the development of automated valuation models in both research studies and real-world online services. Many machine learning models have been reported as the house price estimators to support customer decisions ([1], [2],[3]) Currently, with the emergence and growth of real estate industry, the amount of real estate data has risen annually, impacting many areas such as taxation, obtaining loans from banks, and socialization. The artificial neural network model (ANN) has been widely used recently in this area due to its performance with a large (big) data and ability to treat irregular statistical relations.

An artificial neural network (ANN) is a computational model based on the structure and functions of biological networks of the human brain. The advantage of this model over the hedonic models is its capability to deal with the complex relationships between inputs and outputs such as nonlinear relationships. An ANN is considered to be a powerful data modeling tool based on its ability to represent nonlinear problems and thus allow a broader range of variation. These models can capture different quantitative and qualitative variables that affect the value of the given data ([4], [5]). This means that in the field of real estate valuation, this capability can be very useful in complex systems found in this field where motivations, tastes and budget availability often do not follow rational behaviors. The ANN has demonstrated its robustness as a real estate valuation model comparing to the hedonic models in many cases([5], [6], [7], [8], [9]). Moreover, due to its theory of universal approximation, the ANN is capable of fitting any continuous function, allowing them to capture complex trends, and working with extrapolated data [10]. This make ANN to take advantage over the tree models in real estate area since the house price is possible to be change differently from many factors in a few years.

In spite of the amount of data, the accuracy of this model mostly depends on the amount and completeness of these specific features. This indicates that the different characteristics of each district can influence house prices in different ways. Since the neighborhood characteristics in each larger area, such as the country, are diverse, the local features that affect the house price should be different too. In the area investigated in this study, namely, Bangkok, Thailand, only some residential data have been collected, and the intrinsic influence factors of these data are still unknown. Thus, various local features are collected, considered, and then selected through the feature selection process.

In data science, data are usually represented by high-dimensional feature vectors that in many cases are the key factors for the curse of dimensionality [11]. Data with high-dimensional features will cost more time to treat, consuming greater resources due to their complexity that grows exponentially with dimensionality. The simplest approach to avoid this issue is to perform dimensional reduction to reduce their complexity. On the other hand, it is not easy to determine a priori which features (or input variables) are truly necessary to capture the main characteristics of the studied phenomenon. This is a critical issue, since fast improvements in data acquisition, storage, and management cause the number of redundant and irrelevant features to increase. This leads to reduced learning performance and predictive capability of the models.

Many research studies have illustrated the advantage of a feature selection method over data complexity. Well-known machine learning models, such as ANN or gradient boosting, usually weigh each input feature with its informative to prediction value. Some feature selection methods are applied before a training prediction model to decrease the time consumption and increase model performance. Reducing or filtering noise in the data can also lead to the much-improved efficiency of the prediction model. However, these approaches can be affected by their limitations. Some methods are very effective with a specific prediction model, such as recursive feature elimination (RFE) and random forest, to choose a reasonable

feature selection method for our prediction model. The proper feature selection method will tend to find the best subset of features that obtain the highest accuracy for the specific model with lower time consumption.

In this research, we aimed to address the problem of a limited feature dataset or an enormous dataset with redundant and irrelevant features. Therefore, we proposed a feature selection boosting strategy that can select informative input variables incrementally through each iteration that is suitable for an ANN model. We employ a boosting algorithm to decrease the error rate of the model by focusing on examples that were poorly processed by the previous network [12]. Using this approach, we believe that the resulting weights will lead the estimator to choose a new feature adapted to previously selected features. This method will be useful for applying to real-world regression problems, such as real estate valuation that have a variety of datasets and influenced factors. Furthermore, a new feature sensitivity estimator, called improved Garson algorithm (IGA) [13] was used in this approach to deal with the black-box problem in the ANN models to estimate the importance of each feature in each iteration. Then, the boosting feature selection method with a novel improved approach is provided, evaluated, and compared with other traditional feature selection methods in both a synthetic and real-world data application. Finally, we applied our model to the Kaggle competition dataset, celled the Zillow Prize, to evaluate the performance and rank of our method in this competition.

2. Reviews

In regression and classification problems, a feature selection method is often used to reduce data complexity and noise. These can be divided into the filter, wrapper, and embedded methods. Filters, such as Regressional Relief (RReliefF) ([14], [15]) and correlation-based feature selection ([16], [17]) also called single factor analysis, rank the individual predictive power of each feature or variable according to a specific relevance measurement such as mutual information (MI) ([18], [19]) or Pearson's correlation [20]. Then, the feature with the highest correlation value will be chosen through each method iteration. The wrapper approach, such as RFE [21], uses combinations of features to determine the predictive power and will find the best combination of features through the evaluation criterion of the model. Both approaches can be used with a search strategy to obtain the best result, but these approaches present a major drawback in that they are computationally intractable and time-consuming. Thus, a suboptimal set of relevant features tends to be selected rather than the complete set of useful features. The embedded method is an inbuilt feature selection method that controls the value of the model parameter instead of selecting or rejecting the features or variables and is also called a regularization function. The regularization term is often introduced in the cost function such as in LASSO [22] and RIDGE [23]. These methods are well-suited to treating the problem when the number of potential features is quite restricted. These types of feature selection methods can either work separately by themselves or cooperatively as the ensemble learning based feature selection. the ensemble learning based feature selection is the combination of several feature selection methods and ensemble learning to compensate the inconsistencies between elementary feature selectors and improve the robustness of selection process ([24], [25], [26], [27]) This empirically enhance the selection robustness and overcome the approach with considerable stability improvement in several domains ([28], [29], [30], [31]).

Variable and feature selection have become the focus of many types of research in areas involving the application of large datasets. Domains with large numbers of input variables suffer from the curse of dimensionality in which multivariate methods may overfit the data [32]. A higher number of dimensions theoretically allows more information to be maintained, but practically rarely help due to the higher possibility of noise and redundancy in real-world data [33].

Several studies have demonstrated the potential of feature selection methods to improve predictors in recent years ([34], [35], [36], [37]). Since feature selection aims to reduce the dimension of a dataset by selecting variables that are relevant to the predicting attribute(s), recursive feature elimination (RFE) has been performed to eliminate some of the original input features and retain the minimum subset of features that yield the best classification performance. This method has been either widely composed with other modern models or assembled with other feature selection methods in recent year ([38], [25]) RFE also was illustrated its potential to improve the performance of different types of model in classification task and avoid the overfitting problem [39]. Following the same concept, principal component analysis (PCA) has been applied to feature selection and for selection of several essential individuals from all of the feature components [40].

Another popular choice for feature selection is MI that quantitatively measures both the linear and nonlinear dependence between the variables ([41], [42], [43]). MI is considered as an effective method to select the significant features and deny the undesirable ones by finding the minimum feature subset with the highest discriminative ability that improves model performance. However, MI suffers from the limitations of the parameter distribution, the justified stopping criteria of a greedy search method, and practical usage in a regression problem [43]. For the last

limitation, the adequacy of applying MI as a feature selection criterion has been demonstrated in [44]. Features selected with the MI criterion are the features that minimize the mean squared error (MSE) and the mean absolute error (MAE). By contrast, it was shown that the mutual information criterion fails to select optimal features in some situations. Feature clustering-based methods aiming to find a subset of the features that minimizes the regression error using conditional MI has been proposed ([45], [46], [47]) as an approach for the application of MI in regression problems. These studies show the efficacy of MI to perform in regression task with suitable stopping criteria. This idea was improved further in [48] which proposed a novel stopping condition for MI that can ensure the prediction error boundary.

For regression analysis, several feature selection methods have been applied to increase the predictor accuracy and reduce the computational exhaustion [49].The use of RReliefF, an adapted version of the Relief algorithm [50] for regression problems, has been presented in [14]. Experiments on artificial and real-world data sets show that RReliefF correctly estimated the quality of attributes in various conditions and can be used for nonmyopic learning of the regression trees. An alternative approach has been illustrated in [51]. This approach applies the theorem of the intrinsic dimension to employ a new supervised filter based on the Morisita estimator [52]. It can identify the relevant features and distinguish between redundant and irrelevant information and offers a graphical representation of the results.

In this study, we proposed a new wrapper feature selection method based on ANN and IGA. Moreover, three traditional feature selection methods, RFE, RReliefF, and MI, were considered as a baseline for this study due to their simplicity and efficiency. These methods were applied on the synthetic and real-world datasets,

and their performance characteristics were measured and compared to those obtained with on the ANN model.

3. Background

3.1 Factors Influencing the Housing Price

In recent decades, machining learning methods were involved in real estate area to determine the house prices based on their specific information ([1], [53]) and the trend of price that change over time [54]. The relationship between house prices and environmental factors has been evaluated from various perspectives. A review of the literature showed that many factors influence the house price ([55], [56], [57]), with interest rates, housing construction, unemployment, and household income as important explanatory factors for house prices. This analysis indicated that house prices react quickly and strongly to changes in interest rates. However, this interest rate path reflects an expected decline in unemployment and an expected increase in the growth of wage income. Concerning socioeconomic factors, the population size, percentage of the elderly in population, violent crime rates, and foreclosure rates produced better effects on housing prices in some areas [58]. On the other hand, mortgage rates had less significant effect for housing prices in some areas. Population density, income, and gross value added are considered to be the most significant factors that affect house price fluctuations in London, indicating that population and income are primary indicators of increasing housing prices [59].

House location and neighborhood are also major influential factors that affect house prices. As physically aspect, land-specific topographical characteristic can either increase or decrease the resident value depend on the local topography that give an advantage or disadvantage in each location ([53], [60]). House location also can represent the lifestyle quality of each area that must be considered when purchasing a residence. Proper infrastructure in the housing area also contributes to the increase in house prices. Specifically, there are spatial neighborhood and

location attributes that affect house prices. This variability is more strongly manifested for within-group means than between group means. That is, there are strong variations for individual houses within the same locations and neighborhoods [61]. Various studies on the impact of location and dwelling characteristics on the residential property values/prices were revealed in [62]. The physical and structural specialties of a dwelling, as well as the location of the residential property in terms of accessibility to the workplace, public transportation, proximity to schools, children's playground, and sporting facilities, all contribute immensely to determining the residential property value.

Research carried out in Spain supported the generalization that the location of the housing is an essential factor to consider when purchasing a house [63]. The distance to the central business district (CBD), immigration rate, and socioeconomic factors were proved to be informative factors for house purchasing. Houses that are accessible to the services and facilities and are a short distance from the workplace can help occupants save on transportation expenses. Accessibility has also been proposed to be an influencing factor in [64]. According to urban economics, accessibility improvements resulting from a transportation project influence the residential location choices of households, and the land rents at equilibrium include the valuation of the accessibility gains made by residents. Accessibility to opportunities was found to be a significant factor in property price increases [65]. Furthermore, it is convenient to consider not only the accessibility to transport services, which has been a regular practice in most research but also accessibility to the destination opportunities because it motivates the trips made by users.

Attribute	Description
SalePrice	Property price
Beds	Number of bedrooms
Baths	Number of bathrooms
Parking Lots	Number of parking lots
Size	Total size of the property in sq.m
LotSize	Size of land lots in sq.m.
PropertyType	Types of the property (House, Town house and Condo)
DistrictID	Identifying number of each district in Thailand
SubdistrictID	Identifying number of each subdistrict in each district

Table 1 Description of features in the Home2nd dataset

Attribute	Description
InfRate	Inflation rate (Baht/year)
IntRate	Interest rate (Baht/year).
PurDmnd	Purchase demand (purchashing count/year)
PopGrwRate	Population growing rate in each district (per year).
PopDense	Population density each district (per year).
CrimeRate	Number of crimes that are committed during a month in each district
DistShoppingMall	Distance to nearest shopping mall (km.)
DistRecPlc	Distance to nearest recreation place (km.)
DistGreen	Distance to nearest green zone (km.)
DistEdu	Distance to nearest education center (km.)
DistHealth	Distance to nearest health center (km.)
DistCBD	Distance to Central Business District (km.)
DistMRT	Distance to nearest Metropolitan Rapid Transit station (km.)
DistBTS	Distance to nearest Bangkok Transit System station (km.)
Highway	No. of access point to highway
ImgRate	Immigration rate for each district (per year)
empRate	Employment rate for each district (per year)
AvgIncome	Average income per district (per year)
SaleProp	No. of property for sell in each district

Table 2 *Description of crawled data from OSM*

In this work, we used the real-world dataset obtained from three different data sources: Home2nd from Home Dot Tech Co., Boston housing from the UCI Machine Learning Repository, and Zillow Prize from the Kaggle competition. Every dataset has limited features that contain many missing data depend on the difficulty of the data collecting process in each area. In particular, for Home2nd data, only some residence

information features are available as displayed in **Table 1** that are much lower compared with other datasets since there is no published data collection policy and data center for resale real estate in Thailand. The difference and integrity of the collected data vary due to data sources, possibly giving rise to inaccuracy in property valuation and is not sufficient to make a purchasing decision. As we mention in this section, house prices can be implied in different aspects that inform the user's decision and improve the appraisal model. Hence, we decided to crawl more features from other open source databases, as illustrated in **Table 2**. Although there were many features, the appraisal model was still not efficient enough to use to support a customer's decision due to the characteristic of living conditions in each area. Some features, such as accessibility, are necessary for some areas but will be judged as insignificant features in other areas. Thus, the feature selection method using an ANN was applied to find the intrinsic informative features for a specific field.

3.2 Features important analysis on Artificial Neural Network

The evaluation of input factors in a complex system is both a significant and challenging topic in the sensitivity analysis. An ANN is often viewed as a black box that lead to the limitations in performing factor analysis. Several methods have been defined after a brief description of the connection weights algorithm to quantify the relative importance of independent variables in predicting the output variable for an ANN [66]. Here, we describe the methodologies for analyzing the variable contributions in the ANNs examined in this study.

3.2.1 Garson's Algorithm (GA)

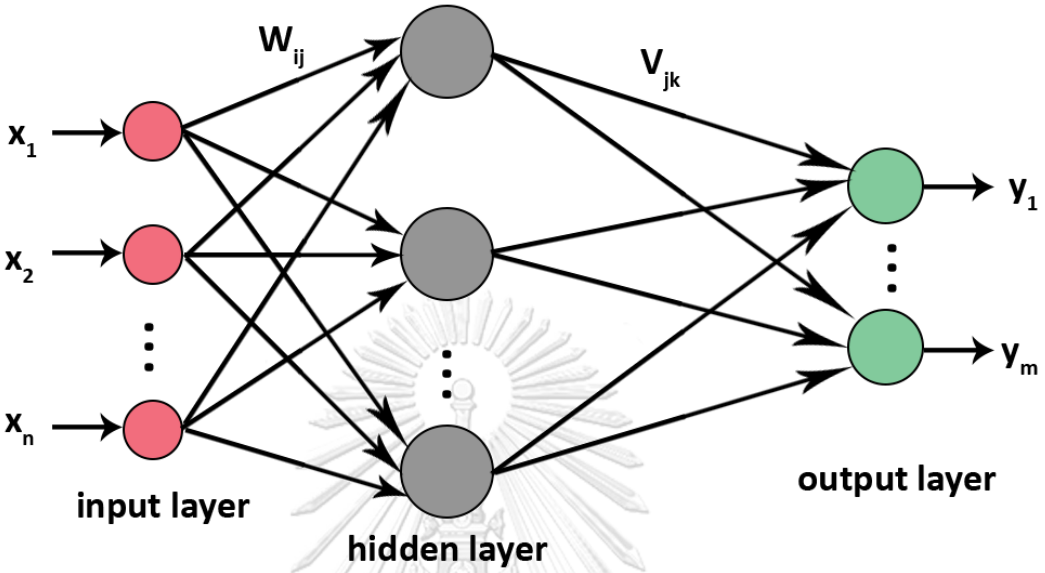

Figure 1 *Structure of a single-layer ANN model used in Garson's algorithm.*

$$S_k^p(i) = \frac{\sum_{j=1}^{n}\left(\frac{w_{ij}v_{jk}}{\sum_{i=1}^{n} w_{jk}}\right)}{\sum_{i=1}^{n}\left(\sum_{j=1}^{n}\left(\frac{w_{ij}v_{jk}}{\sum_{i=1}^{n} w_{kj}}\right)\right)} \quad (1)$$

The IGA is a global sensitivity analysis method that considers the influence of uncertain inputs over the whole input space provided in [13]. This algorithm can solve the stability and precise problem of the original GA. The IGA introduced the system input attribute value x_i, where $i = 1, \ldots, n$, into Garson's output proportional allocation algorithm. The sensitivity coefficient of the \(k\)-th output with respect to the *i*-th input sensitivity coefficient $S_k^p(i)$ was calculated from **Equation 1.**

3.2.2 The Improved Garson's Algorithm (IGA)

The IGA is a global sensitivity analysis method that considers the influence of uncertain inputs over the whole input space provided [13]. This algorithm can solve the stability and precise problem of the original GA. The IGA introduced the system input attribute value x_i, where $i = 1, \dots, n$, into Garson's output proportional allocation algorithm. The sensitivity coefficient of the *k*-th output with respect to the *i*-th input sensitivity coefficient $S_k^p(i)$ was calculated from **Equation 2**:

$$S_k^p(i) = \frac{\frac{x_i w_{i1} v_{1k}}{\sum_{i=1}^{n} x_i w_{i1}} + \cdots + \frac{x_i w_{ij} v_{jk}}{\sum_{i=1}^{n} x_i w_{ij}}}{\sum_{i=1}^{n} \left(\frac{x_i w_{i1} v_{1k}}{\sum_{i=1}^{n} x_i w_{i1}} + \cdots + \frac{x_i w_{ij} v_{jk}}{\sum_{i=1}^{n} x_i w_{ij}} \right)} \quad (2)$$

where p $(p = 1,2, \dots, P)$ represents the sensitivity coefficient calculated with the *p*-th input sample value. The appropriate input attribute sampling method was selected from the entire input space to calculate the sensitivity coefficient. Thus, when the sensitivity of the *i*-th attribute is found, the other input attribute values are also randomly selected in their domain, rather than being fixed as the center value. The IGA is a global sensitivity analysis method and has a greater practical significance. The input attribute selects the appropriate sampling method and sampling points, calculates the sensitivity value for the *k*-th output at each sample point, and then finds the mean $u_k^p(i)$ and variance $v_k^p(i)$, respectively. The sensitivity and interactivity of the input attributes are derived from **Equations 3** and **4**, respectively:

$$u_k(i) = \frac{1}{P} \sum_{p=1}^{P} S_k^p(i) \quad (3)$$

$$v_k^p(i) = \frac{1}{P}\sqrt{\sum_{p=1}^{P}(S_k^p(i) - u_k^p(i))^2} \quad (4)$$

These methods determine the relative importance of the predictor variables of the model as a function of the ANN synaptic weights [67]. The IGA can effectively solve the accuracy issue of GA by considering the influence of uncertain inputs over the whole input space and improve the ranking performance of sensitivity analysis. Furthermore, this method plays its role as a global sensitivity analysis that respects the effect of uncertain inputs over all input space; this approach is more reliable than the local sensitivity analysis method. The application result in the IGA research also illustrated the feasibility and capability for deployment in the data analysis applications. Thus, the IGA was then used to find the optimal feature importance criterion to select informative inputs for the property valuation model.

3.3 Boosting Strategy

Boosting is a method that combines consecutive weak learners to create a stronger learner. In other words, since its goal is to solve the net error from prior networks, the probability of a particular machine depends on the performance of previous learners on that example. This method is usually applied to the decision tree model in the area of a classification problem. In this research, we focused on AdaBoost, one of the well-known boosting strategies.

$$D = max_{i=0}^{n}|e_i| \quad (5)$$

AdaBoost was introduced in 1995 by Freund and Schapire [68] and has become a popular algorithm to iteratively build a classifier as a linear combination of the so-called weak classifiers. At each step, a new weak classifier is added to optimize the classification error rate with a new weighting on training samples. Several methods have been proposed for modifying AdaBoost for regression, such as AdaBoost.RT [69] for use in regression problems that uses the so-called absolute relative error threshold ϕ to project training examples into two classes (poorly and well-predicted samples) by comparing the absolute relative error with the threshold ϕ. Unfortunately, it is, not obvious how the proper value of ϕ should be chosen. Therefore, in this study, we selected another boosting strategy, AdaBoost.R2, [70]. The principal concept of AdaBoost.R2 is very similar to the original AdaBoost, but they differ in the strategy adopted to choose the final prediction.

The method used in AdaBoost.R2 is to express each error concerning the largest error of *n* features, as seen in Equation 5 such that each adjusted error e' is in the range of [0,1]. In particular, one of three possible loss functions is used as shown in **Equations 6-8**:

$$e'_i = \frac{e_i}{D} \tag{6}$$

Or

$$e'_i = \frac{e_i^2}{D^2} \tag{7}$$

Or

$$e'_i = 1 - exp^{(\frac{-e}{D})} \tag{8}$$

Therefore, degree to which instance x_i is reweighed in iteration t depends on how large the error of the hypothesis h_t is on x_i relative to the error on the worst

instance. In this research, apart from the analysis of the impact factors, we considered the effects of the variation in each example that may impact the sensitivity analysis performance. AdaBoost.R2 was deployed with the IGA to improve its performance based on the error in each sample. From preliminary experiments, we also found that this method works consistently well with a square loss function. Thus, AdaBoost.R2 was applied in this study to adjust the probability that the appropriate feature will be selected incrementally in each iteration. The boosting strategy used the error calculated from the previous network with the former set of impact features to adjust the feature selection criterion for the next learner.

4. Proposed Method

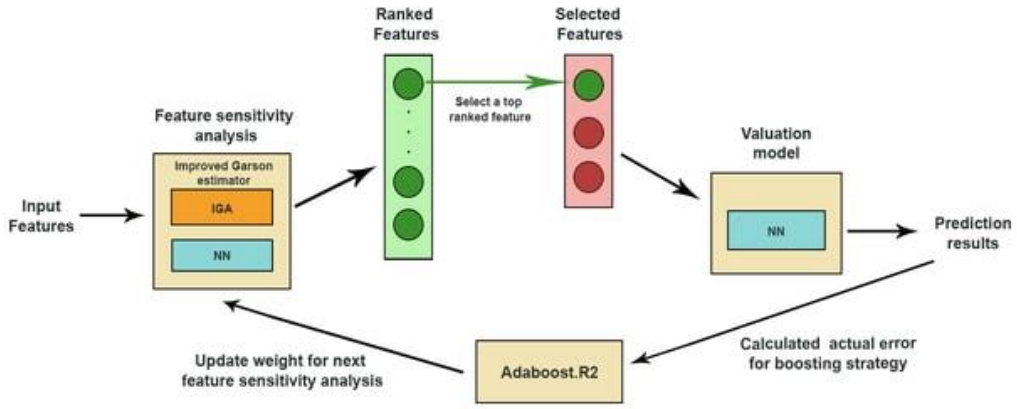

Figure 2 *Overview structure of proposed method.*

In this study, we propose a boosted feature selection strategy that can select the informative input variables incrementally using the Boosted IGA (BIGA) as a selection criterion. The principal hypothesis of this research is that the factors that influenced the house price can be retrieved not only from the effect of feature sensitivity but also from informative data samples. **Figure 2** shows the overall process of the proposed method. The IGA and AdaBoost.R2 are combined to perform as a feature sensitivity estimator. Then, the boosting strategy is deployed to weigh over all the samples via their estimated error. The most relevant factor is then collected through each iteration, then the performance is evaluated with previously selected features until every feature is selected. The feature set that has the lowest error is considered as the set of informative features.

The selected features were estimated by the sensitivity of each feature and the distribution of the error weight in each sample. This method can optimize its selected feature set in each iteration and maintain the character of the IGA as a

global sensitivity analysis. More specifically, the informative features are selected in every iteration without an impact from sample variation.

4.1 The IGA-Based Estimator

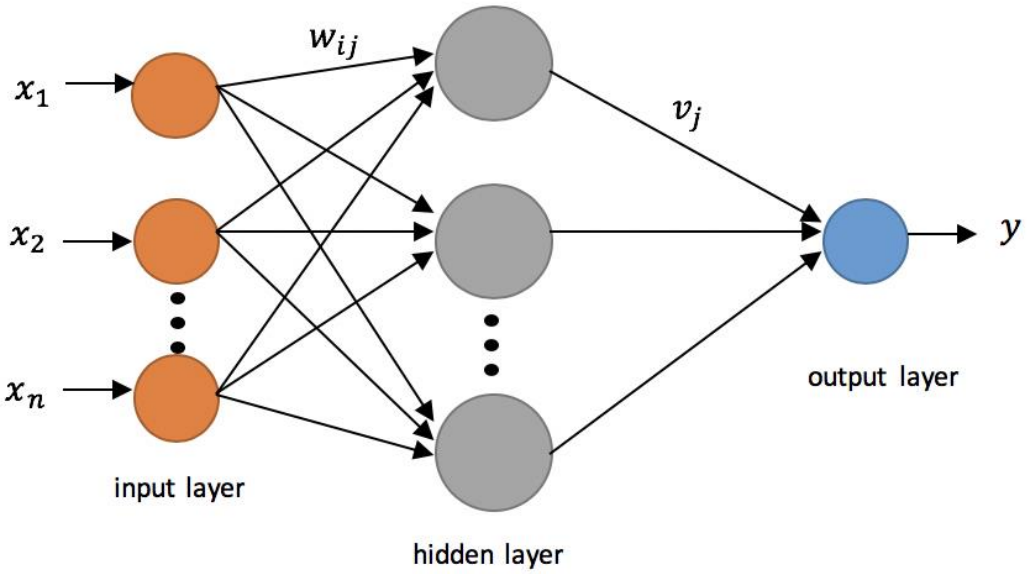

Figure 3 *The structure of 1-layer neural network model used in proposed method.*

In this study, we compared the performance of a single layer and multilayers of hidden neurons structure to select the model structure that obtains the best performance for our method, and then displayed the results in **Figure 3**. The model optimization was performed by a grid search algorithm to find the best topology for each structure. All ANNs were trained using the backpropagation algorithm with the same optimization function. The evaluation method and comparison result are displayed below. Then, we adapted both feature sensitivity importance estimators based on the IGA to be independent from the regression engine and a boosting strategy based on the ANN error.

We have a set of examples X_p, where $p = 1,2, \dots, P$ Each X_p is associated with a value y_p for the result that we want to predict. We divide the data into a training set A and a test set *E*. Then, set *F* of features x_p , $1 \leq i \leq N$ can be computed for each

X_p. The main objective of this method was to select a subset of features $FS \subset F$ adapted to a specific regression model in each iteration $0 \leq t \leq T$. Then, the procedure continues as follows: first, all of the examples were initialized at the same weight $D_p^0 = \frac{1}{P}$, and the set of the selected features was set to be empty. Then, the ANN generated the sensitivity coefficient, S_k^p, that is used in the feature importance estimation for input layer *i*, hidden layer *j*, and output layer *k* with input *p* sample value. Calculation of the sensitivity coefficient was performed as shown in **Equation 9**:

$$S^p(i) = \frac{\frac{x_i w_{i1} v}{\sum_{i=1}^{n} x_i w_{i1}} + \cdots + \frac{x_i w_{ij} v}{\sum_{i=1}^{n} x_i w_{ij}}}{\sum_{i=1}^{n} \left(\frac{x_i w_{i1} v}{\sum_{i=1}^{n} x_i w_{i1}} + \cdots + \frac{x_i w_{ij} v}{\sum_{i=1}^{n} x_i w_{ij}} \right)} \quad (9)$$

where w_{ij} is the connection weights between the input layer and hidden layer neuron, v_{jk} is the connection weights between the hidden layer and output layer neurons. Since our ANN model contains only one output layer, we drop the variable *k* from the equation. After $S^p(i)$ are calculated, the sensitivity and interactivity of the input attributes were ranked using the mean $u(i)$ of each feature sensitivity coefficient. In particular for $u(i)$, we introduced the weighting function on sample, D_p^t, which is a nonnegative function with $\sum D_p^t = 1$. This function is applied here to scale the importance score according to the error of each sample as shown in **Equation 10**:

$$u(i) = \frac{1}{P} \sum_{p=1}^{P} (1 - D_p^t) S^p(i) \quad (10)$$

The result of this estimator is ranked in the descending order of the mean of each feature. The best feature according to this criteria was added to the feature set to be used as a training feature for our the regression model.

4.2 The ANN Model with A Boosting Strategy

After obtaining the selected features in the feature set, the ANN model was applied to generate the predicted results and estimated the error of this set of selected features. The prediction results of the model in this iteration were then used to compute the new weights, D_p^{t+1}, from the previous error provided by previously selected features for each example in the next iteration using the AdaBoost.R2 algorithm.

The ANN regressor was applied with the selected data in the feature set to generate the hypothesis $h_t: X \rightarrow R$ in each iteration *t* with the set of selected features from the input variable *X*. Each adjusted error e'^t_i was calculated by mapping each prediction error into an e'^t_i for each instance, to express each prediction error in relation to the largest error, using **Equations 11 and 12**:

$$E_t = max_{i=1}^{p} |y_i - h_t(X_i)| \quad \textbf{(11)}$$

$$e'^t_i = \frac{(y_i - h_t(x_i))^2}{E_t^2} \quad \textbf{(12)}$$

Then, this function was averaged over all of the weighted examples to calculate the adjusted error of h_t as shown in **Equation 13**:

$$\epsilon_t = \sum_{i=1}^{P} e'^t_i D_p^t \quad \textbf{(13)}$$

Knowing the adjusted error of the hypothesis, ϵ_t, the weight updating parameter (denoted β_t) was computed using **Equation 14**:

$$\beta_t = \frac{\epsilon_t}{(1 - \epsilon_t)} \quad \textbf{(14)}$$

Finally, we used the sample weighting function in the previous network to calculate the weighting function for the next feature selection criterion in the next iteration after updating with the updating parameter, as shown in **Equation 15**:

$$D_p^{t+1} = \frac{D_p^t \beta_t^{1-e_p^t}}{Z_t} \quad \textbf{(15)}$$

where Z_t is a normalizing constant.

The weight of each sample was updated since it is a misleading result. The samples with a larger error will receive the higher weights than the others. By doing this, the next resulting weights will improve the feature selection criteria and lead it to choose a better feature.

The weight of each sample was updated since it is a misleading result. The instance with a large error receives a higher weight than the others. By doing this, the next resulting weights will improve the feature selection criteria and lead it to choose a better feature.

5. Experimantal Study

5.1 Experimental Setup

To test our approach, one synthetic dataset and three real-world datasets were used with the proposed method. The synthetic data was used as a feature selection performance indicator because its important features were significantly set from the beginning. Then, the real-world data were processed in a feature engineering process and then used to estimate the capability of the proposed method to select essential factors in the real-world property dataset.

5.1.1 Friedman Dataset

This dataset was used to validate the multivariate adaptive regression splines (MARS) models to uncover the structure in the data. The output Y is given by **Equation 16**:

$$Y = 10\sin(\pi X_1 X_2) + 20(X_3 - 0.5)^2 + 10X_4 + 5X_5 + \epsilon \qquad \textbf{(16)}$$

where X_1, X_2, X_3, X_4 and X_5 are independent and equally distributed variables and ϵ is Gaussian random noise with zero mean and unit variance. The input space of this function has three nonlinear and interacting, along with two linear variables, and five that are irrelevant. Finally, the Friedman dataset is produced by randomly sampling *N* points from the inputs. In this research, the sample size was $N = 1000$ points.

The Friedman dataset is characterized by a nonlinear structure, and its input space contains extra variables that can be removed without affecting the learning of the target Y. For this reason, we chose this dataset to validate the ability to select

the relevant variables and to remove the irrelevant and redundant variables of our method.

5.1.2 Boston Housing

Type	Attribute
CRIM	Crime rate per town.
ZN	Proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS	Proportion of non-retail business acres per town.
CHAS	Charles River dummy variable (1 if tract bounds river; 0 otherwise).
NOX	Nitric oxides concentration (parts per 10 million).
RM	Average number of rooms per dwelling.
AGE	Proportion of owner-occupied units built prior to 1940.
DIS	Weighted distances to five Boston employment centres.
RAD	Index of accessibility to radial highways.
TAX	Full-value property-tax rate per $10,000.
PTRATIO	Pupil-teacher ratio by town.
B	$1000 * (Bk - 0.63)^2$ where Bk is the proportion of blacks by town.
LSTAT	Lower status percentage of the population.
MEDV	Median value of owner-occupied homes in $1000's.

Table 3 *Description of features in Boston Housing dataset*

This dataset, created by Harrison et al. is derived from the information collected by the U.S. Census Service concerning housing in Boston, Massachusetts. The objective was to predict the housing prices in different areas of Boston. The dataset contains 506 instances and 14 continuous features and can be found in the UCI Machine Learning Repository. The description of each variable is illustrated in **Table 3**.

For this dataset, we used the interquartile rule for outliers to identify the outliers in the data. The interquartile range (IQR) was calculated by subtracting the first quartile from the third quartile to demonstrate how the data is spread around the median. Then, the IQR was multiplied by 1.5 according to the rule, and then the data samples that were out of the IQR range were eliminated.

5.1.3 Zillow Prize

This dataset is provided in the Kaggle competition by Zillow. The main objective of this competition is to predict the logerror between their valuation model prediction calculated from Equation 17, and the actual sale price for the months in Fall 2016 and 2017.

$$logerror = \log(Zestimate) - \log(SalePrice) \quad (17)$$

The complete dataset consisted of 2,985,217 cases with 58 features including property information (ex. air condition type, living space, number of rooms), location (ex. latitude, longitude, neighborhood area), and taxation. The competitors have to create training and validation data from 90,275 and 77,613 transaction points in 2016 and 2017, respectively.

For preprocessing, we eliminated the outlier data using the same IQR method as that used in the previous dataset. Then, the features that contain more than 98% of the missing value or contain only one unique value were classified as uninformative features and dropped. Finally, these features were separated into two groups according to their type: numerical and categorical. Numerical features were employed in the feature extraction process to produce location-based and neighborhood-based attributes and then imputed the missing values with the mean of each feature. For categorical features, we imputed the missing values with one negative value and represented their value in numeric data through the embedding layers in model.

5.1.4 Home2nd

The data provided at www.home2nd.com contains resale real estate information in Thailand. The dataset contains data for 43,922 properties in Bangkok and 63 features, including house information, location information, date-time, post information, and house price (including both sale price and rental prices.) In this study, we considered only 14 out of all the features that could impact house values and focused on only the sale price that was provided in terms of a price range. We decided to focus only on the property information and location features discussed in the previous section.

According to [71], many other kinds of features can be beneficial for obtaining an accurate price range. Therefore, the data crawling process from OpenStreetMap (OSM) was carried out to acquire the data of the surrounding environment. The OSM is a free, editable map of the whole world that is being built by volunteers from scratch and released with an open-content license. The points of interest data, such as amenity places, green zone, and recreation place, and transportation stations were obtained through the database web API called Nominatim. Nominatim is an

open-source geocoding associated with OSM data. This API was used to calculate the distance to each nearest point of interest (POI) that we illustrated in the prior section. We also extracted some more features from unselected features that we expected to be an informative feature in our valuation model, such as days on market (obtained by counting the days between the CreateDate and CloseDate variables). Furthermore, location features, such as the latitude and longitude values, were used as the basic information to crawl the neighborhood environment data of each property from the OSM. We narrowed down the focused area to be only in Bangkok. Thus, the features that have a larger scale than the district area were not considered, and this reduced the number of samples to 11,697 including three types of property in Bangkok (Condo, Home, and Town House). As a result, these features from Home2nd and OSM were combined to construct the new Home2nd dataset used in this research.

The Home2nd data were processed to be suitable for feature engineering and then imputed for their missing values by obeying the following conditions: for every quantitative continuous data, such as the property and distance from the POIs, we filled them with the mean value for each feature and mean value in each district for those that were related to the location. For every quantitative discrete feature, we filled them with the median value. Finally, the outliers in the data were also identified using the IQR.

5.2 Evaluation Setup

We evaluated model performance with two estimators, namely, the root mean squared error (RMSE) and the mean arctangent absolute error (MAAPE) to obtain the validation feature set. Furthermore, the predicted results from the Zillow dataset

were also evaluated on Kaggle.com using mean absolute error (MAE) to be ranked in the leaderboard.

5.2.1 Root Mean Square Error – RMSE

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y'_i - y_i)^2}{n}} \quad (18)$$

The RMSE was used to measure the standard deviation of the residuals that are a measure of how far from the regression line data points are; RMSE is a measure of how to spread out these residuals are, calculated by Equation 18; where y 0 is predicted value, y is the observed value, and n is the total number of samples.

This estimator is a good measure of accuracy, but only for comparing prediction errors of different models or model configurations for a particular variable and not between variables because it is scale dependent.

5.2.2 Mean Arctangent Absolute Percentage Error – MAAPE

$$MAAPE = \frac{1}{N}\sum_{i=1}^{N}(\tan^{-1}(\left|\frac{A_i - F_i}{A_i}\right|)) \quad (19)$$

for $i = 1, \ldots, N$

MAAPE is a new estimation method developed from the well-known estimator called mean absolute percentage error (MAPE) proposed in [72]. The MAAPE can overcome the MAPE limitation of going to infinity as the actual value goes to zero by bounding its range with the arctangent as shown in **Equation 19**; where A and F are the actual and forecast values, respectively.

The MAAPE obtains a more balanced penalty between positive and negative errors than MAPE, although the penalty function of MAAPE remains asymmetric. This estimator can also be particularly useful when enormous errors are due to mistaken or incorrect observations. We decided to use this estimator to express the performance of the model without the impact of the scale of the product. Moreover, it is quite easy to explain the error of the results in a business-related manner.

5.2.3 Mean Absolute Error

$$MAE = \frac{1}{N}\sum_{i=1}^{N}(|y'_i - y_i|) \quad \textbf{(20)}$$

MAE is the mean of the absolute values of the individual prediction errors on over all instances in the test set. It was used to measures the average magnitude of the errors in a set of predictions, without considering their direction. The calculation is shown in **Equation 20**; where y' is predicted value, y is the observed value, and n is the total number of samples.

In this study, MAE was applied as an evaluation metric in Zillow Prize competition. Submissions are evaluated on MAE between the predicted log error and the actual log error to ensure that valuation models are not biased towards expensive homes.

Since this study is based on an ANN where the hyperparameters can affect the model performance, the hyperparameters of that ANN were considered as one of the major keys in this study. These variables were set differently according to the scale of features and samples in each data. We ran an experiment to find the most

suitable ANN structure for every data. The parameters of the network model were set as follows: the rectified linear unit (ReLU) was applied as an activation function for both weights learning between the input layer and hidden and between the hidden layer and output layer. The learning function used Adam with a learning rate of 0.001, performance function as MSE, and the remaining parameters were set as default. For Zillow, we assign an ANN model to be more complex to produce the prediction result that could be placed in a higher rank. First, the embedding layers were employed for each categorical feature to represent the discrete value into the continuous vector. Then, the inputs from the embedding layers were concatenated with the inputs from a numerical input layer before feeding into the hidden layers. This model contained three hidden layers in which their activation function was set as a one tanh function, and two ReLU function, respectively. Every hidden layer was connected to the dropout layer to prevent the results from the overfitting problem. We also used Adam in this model, with a learning rate of 0.005. The training and validation in each data were randomly selected five times through the cross-validation method with the same model structure as used for model performance validation.

5.3 Result and Discussion

Dataset	Single-layer		3-layers		5-layers		7-layers	
	MAAPE	RMSE	MAAPE	RMSE	MAAPE	RMSE	MAAPE	RMSE
Friedman	15.31221	2.553	13.715	2.308	15.868	2.622	16.422	2.775
Boston	28.917	9.181	14.561	4.219	14.923	4.461	15.156	4.488
Home2nd	41.7268	0.5421	31.7525	0.4189	28.953	0.3776	30.063	0.443

Table 4 *Comparison result between the three model structures for the ANN model. Every model is trained using all of the features of each dataset in the regression task.*

First, we performed an ANN structure comparison to find the model that yields the best performance for our method. Since the topology of an artificial neural network is determined by its structure and hyper-parameters, the different single-layer and multilayers ANNs model were constructed through grid search algorithm, and then we perform a task in each dataset to obtain a comparison score. The result from each task was evaluated by the RMSE and MAAPE estimators as displayed in **Table 4.**

An examination of the results shows that more complex structure of ANN can perform better than a single-layer model, but not in all cases. For the Friedman and Boston datasets that are small-sized datasets, the 3-layer ANN obtained lower RMSE and MAAPE scores than the more complex one, but in a medium-sized dataset such as Home2nd, the 5-layers obtained the lower error instead. This illustrated the effect of data size on the model structure and performance. An inadequate match between model complexity and data size can cause problems in the model learning process such as underfitting and overfitting. In this experiment, the average samples of the benchmark dataset are approximately 5000-20,000 samples. Therefore, we decided to use a medium complexity structure, a 3-layer ANN model, as a baseline model for our method since it can perform much better than a single-layer ANN and

its performance is not much different from the more complex model according to the data size.

Then, to indicate the method behavior, we compared the proposed method with the well-known feature selection techniques of RFE, RReliefF, and MI.

The RFE is a feature selection method that fits a model and removes the weakest feature (or features) until the specified number of features is reached. The features are ranked by their coefficient and importance score, and by recursively eliminating a small number of features per loop. Thus, the RFE attempts to eliminate dependencies and collinearity that may exist in the model. Although the RFE requires maintaining a specified number of features, it is often not known in advance how many features are valid. We decided to add the best feature in the ranking feature and calculate the new feature at each iteration. The set of features that provided the least error was then considered as the final result of this method.

As stated before, RReliefF is an adaptation of Relief and ReliefF to regression. In regression, the exact knowledge of whether or not two instances belong to the same class cannot be used. RReliefF replaces this with a probability that the predicted values of the two variables will be different and computes the final score of each feature by considering kRF neighbors. Among these neighbors, the closest neighbors should have a significant influence, and a parameter kernel, , can be used to assign a weight to each of them.

The last baseline method for this study is MI. The basis of this method is to apply the information gain as the feature selection estimator. This method selects the

features based on a measure of the amount of the reduction in the uncertainty for a variable given the known values of the other variable. This definition is useful within the context of feature selection because it provides an approach for quantifying the relevance of a feature subset with respect to the output vector. A feature subset with a high mutual information with the target output is likely to reduce the uncertainty on the values taken by the output and were selected to be a member of informative features. The MI method used in this experiment was from the extensively use Python library, scikit-learn, which was based on the improved version of MI for discrete and continuous data [46].

In this experiment, every feature selection technique was carried out as a forward selection method, and then the feature that best improved the model were added until every such feature was selected. The set of selected features in each iteration were evaluated using root mean square error (RMSE) values for the Friedman, Boston housing, and Home2nd datasets. Each method evaluated the results in each iteration was illustrated for each associated dataset. Then, the comparison result among all feature selection method was operated and discussed. Finally, we applied these methods with other general regression dataset to provide the benchmark in generalization perspective and illustrated the performance of our model in different cases of data perspective.

5.3.1 Synthetic Data

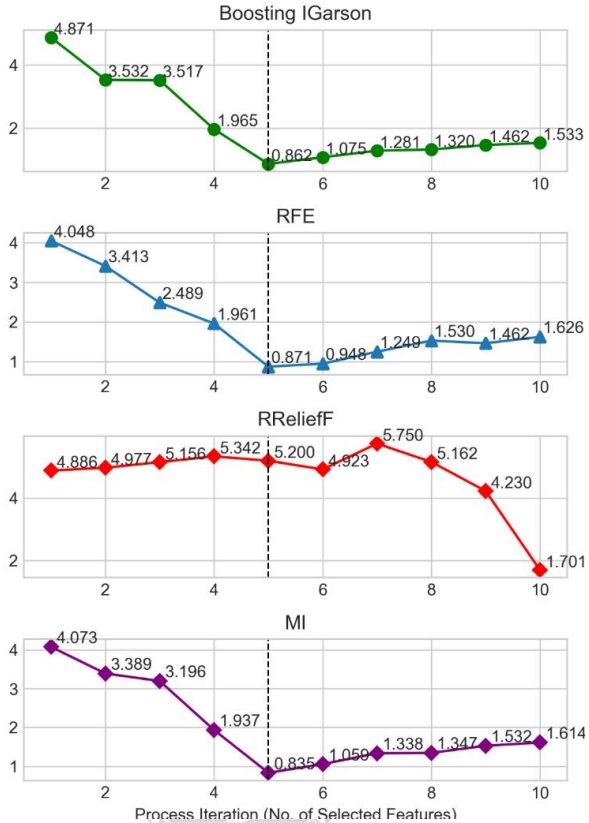

Figure 4 *Comparison among three algorithms on the Friedman dataset.*

We approached every method using synthetic data. **Figure 4** shows the evaluated result for each set of selected features in each iteration using the Friedman dataset. The proposed method of this study could significantly select each informative features in each iteration until they were completed. From the different decreasing error rates in each iteration of each algorithm, RFE and MI start from lower error than other methods, implying that these methods selected features that highly reduced the error first, as the others determined a feature that reduced less error. The BIGA provided an overall perspective on the influence of the inputs on the outputs, similar to the local perspective of partial derivatives in the RFE and MI. Thus, our method will not select features that highly reduce the model error at first but maintains a more important score for informative features until they are selected. By

contrast, RReliefF appears not to work well in this case since it fails to reduce the error after the first **iteration.**

5.3.2 Real-world Data

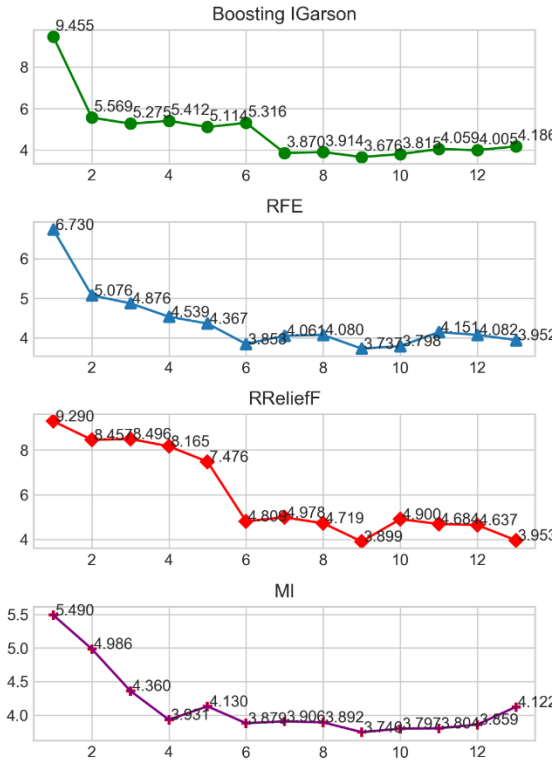

Figure 5 Comparison among four algorithms on Boston housing data.

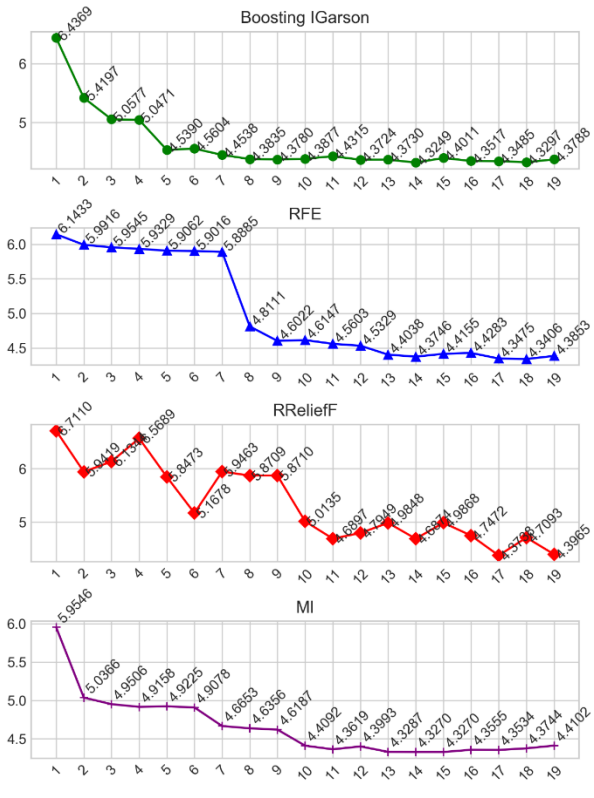

Figure 6 *Comparison among three methods on the Home2nd dataset.*

Then, every feature selection algorithm was applied to measure and select the informative variables in the real-world data using the Boston Housing and Home2nd datasets. Backpropagation Neural Network was used for modeling the input and output data sets of the system. The model network adopted a three-layered structure, comprised of the input, hidden, and output layers. The model parameters were set to be the same as in the previous model. The results of each algorithm are shown in **Figure 5 and 6**. According to the results, our method, RFE, and MI continuously reduce the model error by considering the informative features in most iterations. Their graph have approximately the same slope and are clearly different from RReliefF in the early iterations. Our method finishes with the same number of selected features as MI (9 for Boston and 14 for Home2nd) and shows a slightly lower RMSE score. This demonstrated that our method can precisely select the informative feature according to the improved error from boosting strategy and

perform slightly better than MI and other methods in these cases. Moreover, this experiment implied that increasing the number of features does not signify a stable improvement of model accuracy. Similarly, decreasing the number of features too much will cause a high and significant error due to the problem of a lack of information.

	Feature selection method	MAAPE	RMSE	No. of features
Friedman	Unselected	13.442	2.231	10
	BIGA	4.8788	0.861	5
	RFE	4.906	0.871	5
	RReliefF	9.412	1.583	10
	MI	**4.8436**	**0.835**	5
Boston Housing	Unselected	14.887	4.337	13
	BIGA	**12.966**	**3.673**	9
	RFE	12.691	3.737	9
	RReliefF	13.374	3.899	9
	MI	13.020	3.745	9
Home2nd	Unselected	35.187	0.533	19
	BIGA	**29.618**	**0.432**	14
	RFE	30.799	0.434	18
	RReliefF	33.801	0.469	17
	MI	30.615	0.432	14

Table 5 *Estimated computing error of the model with all of the features and selected features from BIGA, RFE and RReliefF*

Finally, the comparison results of the best score in each technique for every dataset are illustrated in **Table 5**. The RMSE and MAAPE were used to evaluate and compare the performances of the methods. The results presented in **Table 5** indicate that our method produced more promising results with equal number or fewer selected features. The superior results are shown in bold, where the procedure of this study gives better results with fewer selected features than other methods in Boston Housing and Home2nd dataset. For the Friedman dataset, MI show the best performance in this case while RFE and our method have approximate error since MI obtained the information of two random variables by observing their entropy without

the effect from model performance or prediction result while RFE and BIGA utilize wrapper feature selection method and that their performances are affected by the based neural network model with small dataset. This indicated the effect of model complexity and data size to method efficiency. Inadequate model and data size can reduce BIGA and RFE performance because it uses the error from the prediction result to calculate the feature importance score. By contrast, as observed in the case of the Boston Housing and Home2nd, our method showed a slightly more stable performance in selecting the informative features incrementally in each iteration. There is quite amount of strong and close to strong correlations happening among feature variables in Boston Housing dataset that can lead to skewed or misleading results for predicting housing price when using each of them separately. Although the data size is so small (506 samples), BIGA and RFE can present its capability for dealing with multicollinearity and keep precisely selecting an informative feature. BIGA and RFE outperforms MI in this case because of their capability to handle the multicollinearity while MI is unable to detect such correlation. What MI does is to consider only the entropy between a target and one random feature. In the case of Home2nd, there are much greater number of samples than Boston Housing and Friedman dataset that is suitable for neural network model. Hence, BIGA works rather well for it is an ANN based, and could outperform MI in this case. Nonetheless, although BIGA outperforms MI, the gap is small due to the weak correlation among feature variables and the low impact of each feature. Despite the result, our method keeps selecting the right feature until it reached the optimal number of features that is less than the selected features from RFE with less error.

In the case of Home2nd, we also performed the data segmentation experiment. The data in Home2nd was separated into 3 set by property types (home, condo, and town house). The distribution of each data was illustrated in **Figure 7**.

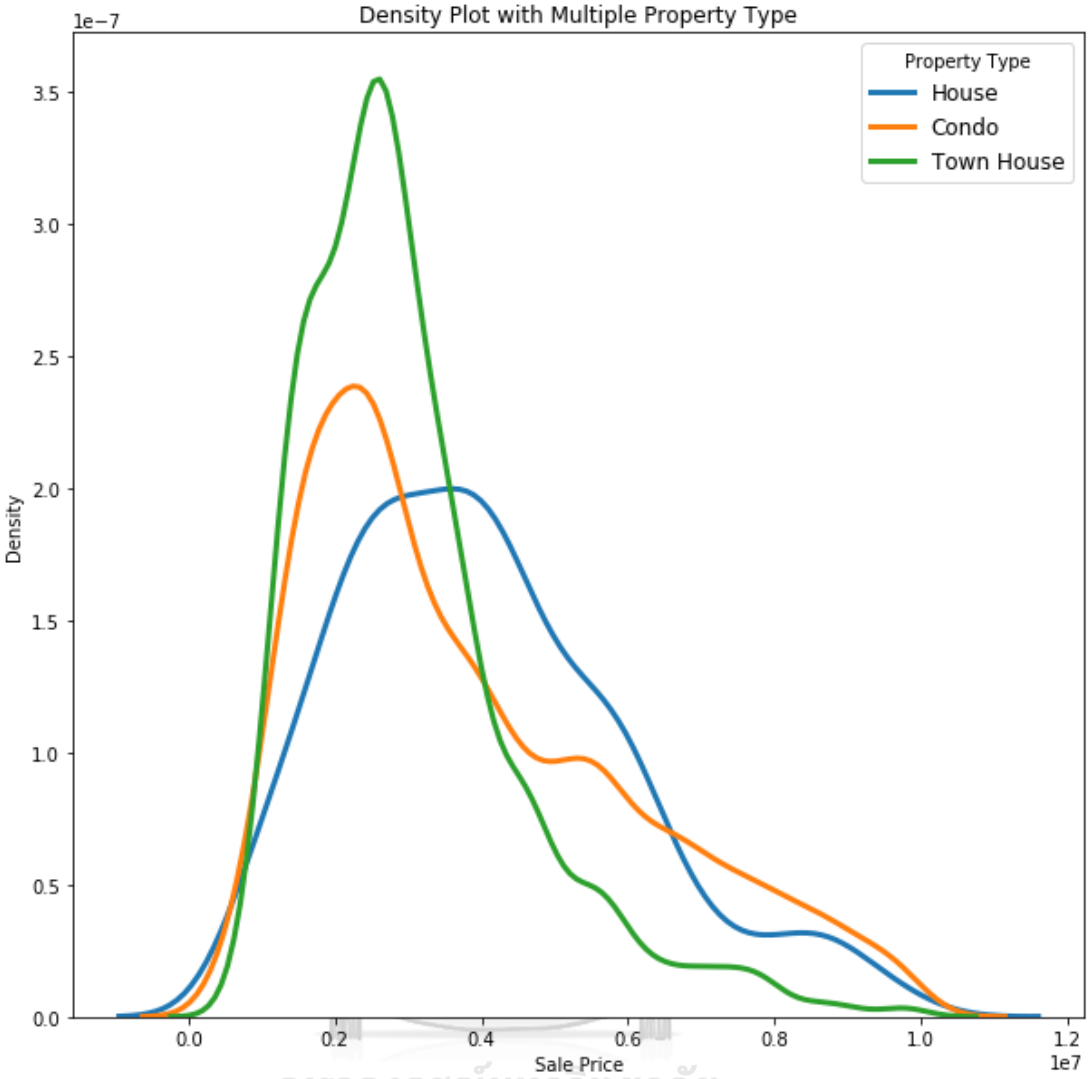

Figure 7 *Data distribution of each dataset in Home2nd separated by property type.*

Property Type	Method	MAAPE	RMSE
Home	Unselected	33.2322	0.5076
	BIGA	29.5065	0.412
	RFE	30.0523	0.4209
	RReliefF	30.5179	0.4306
	MI	29.8717	0.4481
Condo	Unselected	25.85466	0.42753
	BIGA	24.3502	0.42
	RFE	24.7556	0.4245
	RReliefF	24.8309	0.4175
	MI	24.3513	0.42332
Town House	Unselected	25.67	0.3907
	BIGA	24.5501	0.3759
	RFE	24.9311	0.3861
	RReliefF	25.0563	0.3796
	MI	23.7551	0.3675

Table 6 *The result from data segmentation experiment.*

Result of this experiment was shown in **Table** 6. BIGA can highly reduce the overall error in case of home data that has normal distribution but lessen its performance in condo data which has skewed distribution. This indicated that BIGA was affected by the data distribution. However, in town house data, MI had the best overall error score that outperformed other methods in this case even though data in this case has approximately normal distribution. This showed the effect of amount of data over the data distribution problem in BIGA since town house data has the least data sample among every data segment (around 2500 samples).

Moreover, we employed our method with the Zillow Prize data using the evaluation method of the Kaggle competition. This competition used MAE to measure the average magnitude of the errors in a set of prediction results, without considering their direction. It is the average over the test sample of the absolute

differences between the predicted and actual observations where all individual differences have equal weight. Kaggle automatically calculated the MAE between our predictions and the actual values and then ranked our results based on the computed MAE score on the leaderboard. We started by comparing our method and the base line methods with Zillow data on a 3-layer ANN and then validated by 4th quarter data. We applied the best selected features from each method as the training features for this competition. The evaluation results were illustrated in **Table 7.**

Model	MAE	Rank	Rank%
ANN	0.06490	2398	64
BIGA	0.06439	1027	24
RFE	0.06467	2037	54
RReliefF	0.06488	2340	62
MI	0.06463	1689	44

Table 7 *Comparison result between feature selection methods in Zillow competition.*

From the result, BIGA show the most improvement for ANN model and yield the best score and rank among every method. The predicted results from only ANN model earn a 0.06490 score on the public leaderboard that was ranked in approximately the top 64% of over 3700 competitors. This result was as expected because the Zillow dataset contain many categories of data such as the zip code and the county land use codes that have a large number of unique elements. These features result in an enormous number of weight vectors, most of which are irrelevant. BIGA earned a 0.06439 MAE score and ranked in the top 24% of the leaderboard. For comparison, we also submitted the result from two tree-based models that widely used in this competition, Extreme Gradient Boosting (XGB) and Category Boosting (CatBoost). The result was shown in **Table 8**.

Model	MAE	Rank	Rank%
ANN+BIGA	0.06439	1027	24
CatBoost	0.06433	585	15.5
XGB	0.06454	1512	40

Table 8 *Comparison result between three models in Zillow competition.*

Our method was ranked in the higher position than XGB. This can be explained by the limitation of tree model. XGB is unable to extrapolate target values beyond the limits of the training data due to the method with which tree-based models partition the input space of any given problem. Zillow used transaction in 2016 and the 1st quarter in 2017 as training data, and other 3 quarters in 2017 as testing data. There may be some extrapolated data occur in last quarter in 2017 that is not familiar with XGB. In contrast, ANN is a proper model to capture an increasing trend and predict values outside the range of the training data. In addition, XGB deal with category data with one-hot encoding which make XGB get bad result. Our method can also outperform the traditional models and was close to CatBoost which was considered to be the proper model for this competition. This experiment shows the capability of ANN to be used as valuation model and our method to indicate high potential to improve the performance of original ANN model.

The objective of this participation is to characterize the efficiency of our method to improve ANNs' performance, and, from the result, it showed the high potential about this.

5.3.3 General Case

Datasets	Instances	Cat Attr.	Num Attr.
Insurance	1338	3	3
SeoulBike	8760	3	10
WellLog	8000	0	13
KCHouse	21613	0	18
OLSReg	3047	0	31

Table 9 *Brief description of the general regression datasets*

In this section, we applied our method and baseline methods with other regression dataset to use as a benchmark for comparison with other baseline models and simulate the worst case and the best case of our method. For this purpose, we used 5 datasets for the regression problems from the UCI repository [73] and Kaggle. **Table 9** provides a brief description of these data sets.

	Feature selection method	MAAPE	RMSE	No. of features
Insurance	Unselected	33.549	0.409	6
	BIGA	29.667	0.382	5
	RFE	29.398	0.361	4
	RReliefF	33.576	0.422	6
	MI	32.199	0.536	5
SeoulBike	Unselected	41.022	0.383	12
	BIGA	36.729	0.356	10
	RFE	38.803	0.380	10
	RReliefF	39.637	0.381	11
	MI	38.099	0.378	10
WellLog	Unselected	21.198	0.295	13
	BIGA	19.276	0.268	12
	RFE	19.927	0.274	12
	RReliefF	19.530	0.274	11
	MI	18.442	0.262	12
KCHouse	Unselected	14.968	0.268	18
	BIGA	13.507	0.249	16
	RFE	13.757	0.252	17
	RReliefF	14.041	0.250	17
	MI	14.059	0.300	15
OLSReg	Unselected	9.238	0.121	31
	BIGA	8.602	0.116	28
	RFE	8.573	0.115	29
	RReliefF	8.887	0.120	28
	MI	8.586	0.116	29

Table 10 *Estimated computing error of each method with the general regression datasets*

We present the experimental results of each method in **Table 10**. We use the same model structure and parameters as those used in the previous section based on the ANN model. MAAPE and RMSE are used as the evaluation methods to compare the performance characteristics of each method. The number of features is the number of features selected in the best iteration of each method.

An examination of the results shows that every method reduced the overall error for each dataset. This implied the important role of feature selection method in the different areas of data. Among all of the methods, BIGA can perform better in almost medium-sized data (SeoulBike, WellLog, and KCHouse) with the same number or fewer features. By contrast, for small-sized dataset (Insurance and OLSReg), RFE and MI are preferable. This indicated the effect of the inadequate model structure and data size on model performance. Since BIGA is based on ANN performance, the

precision of this method can be affected by model performance. The unsuitability of model structure and data size will reduce the BIGAs' efficiency in the selection of the proper features. RFE, RReliefF, and MI were not affected by the same problem because they were calculated separately from the base model. However, BIGA still has potential to select some the informative features and reached the highest performance when model is not too complex and has enough training data.

In conclusion, BIGA is used to analyze the impact factors in various datasets. This method demonstrates its capability to evaluate the sensitivity for each feature over the entire range of each input factor. As expected, BIGA can choose a feature that reduces the overall error through each iteration until it reached the lowest error according to the practicability of the model structure and data size. This result indicates the effectiveness of the boosting strategy for improving feature selection criteria based on previous prediction error and representing the impact of input variation. The selected features can be considered as an intrinsic informative feature for the dataset and implied the compatibility of this model to analyze the features that influence house prices. However, BIGA consumes more resources and processing time depending on the scale of the dataset since it is measured based on the input features and sample weights. The stopping criteria are also an essential issue for this method, as for the other feature selection methods. The BIGA consumes the surplus by continuously selecting a feature until every feature was selected even though every informative feature was picked. This process cannot be stopped during the selection process because there is no precise estimator function that is suitable for the BIGA since it does not choose the most informative feature (a feature that reduces the most overall error) in each iteration. However, this experiment shows that BIGA is suitable for improving the model accuracy by reducing the complexity of the data and avoiding the effect of the curse of dimensionality, and it can identify the intrinsic factors for real estate valuation.

5.3.4 Runtime Analysis

Source	Dataset	Time(ms)					
		Only IGA	IGA Only ANN	Total	RFE	RLF	MI
CPU	Friedman	40	12200	12250	5.1	6130	114
	Boston	67.5	4640	6704	1.6	4010	69.7
	Home2nd	3429	23700	27130	3.93	123000	5260
GPU	Friedman	30.5	125	160.5	-	-	-
	Boston	20.29	48.9	73.7	-	-	-
	Home2nd	187.7	226	467	-	-	-

Table 11 *Runtime analysis result on training process*

Source	Dataset	Time(ms)				
		BIGA	RFE	RLF	MI	Unselected
CPU	Friedman	2769	2838	3401	2812	7063
	Boston	10936	13721	17029	12393	21554
	Home2nd	2016	5102	7628	2806	10315
GPU	Friedman	66.7	68.2	69.7	63.4	70.1
	Boston	61.8	62.6	63.4	63.4	83.6
	Home2nd	144.7	154.9	155.1	152.9	200.7

Table 12 *Runtime analysis result on prediction process*

We also did the runtime analysis of each method on Friedman, Boston housing, and Home2nd dataset. We compared BIGA without boosting strategy with other baseline methods to measure the time that these methods consumed as feature selection methods in training and differentiate the prediction time to measure the inference performance. This analysis was done on both CPU and GPU, and then displayed in **Table 11 and Table 12**

6. Further Study

Based on the experiments, we will examine the feasibility of modifying BIGA to choose the most informative feature in each iteration or adapting other methods with a boosting strategy that is more effective. Stopping criteria is an addition interesting issue to lessen the computing time of our method. Moreover, we are also interested in the application of our study in the ensemble feature selection method and the valuation system to support customer's decisions and inspect the characteristics of the real estate industry from the selected features in global scale.

Appendix A: Mathematical Model

A.1 RFE

Recursive feature elimination (RFE) is a wrapper-type feature selection algorithm. This means that a different machine learning algorithm is given and used in the core of the method, is wrapped by RFE, and used to help select features. RFE works by searching for a subset of features by starting with all features in the training dataset and successfully removing the least importance feature until the desired number remains. The feature importance is considered by the feature coefficients which are the same as the coefficients we get after fitting the model on dataset after minimizing the residuals. The overall process for selecting features using the feature-importance-based RFE method is shown in **Figure 8** below.

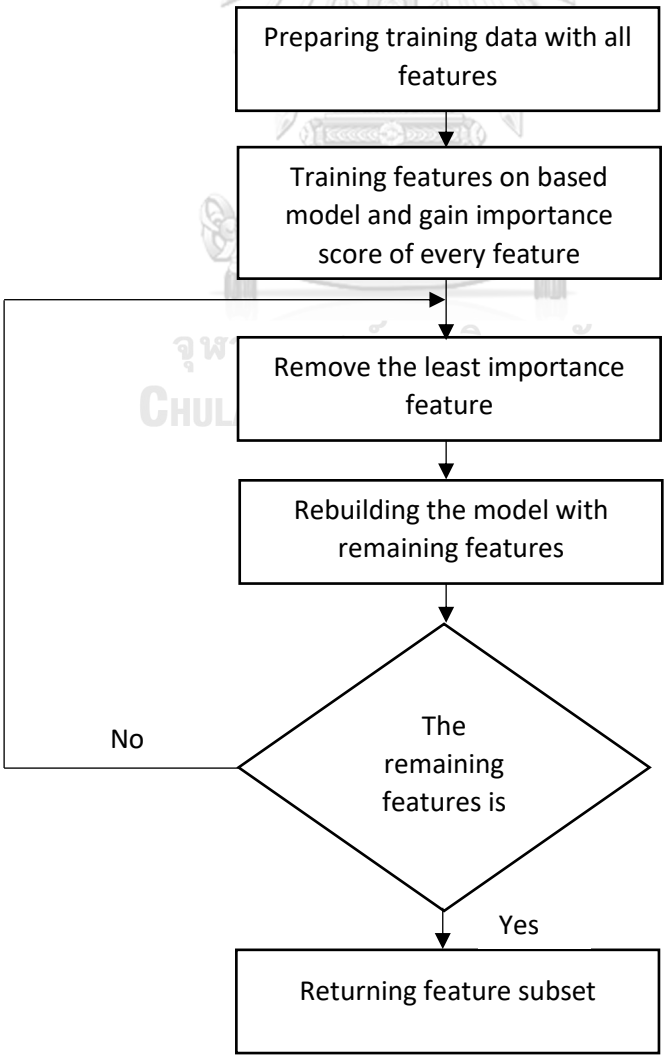

Figure 8 *Overall process of RFE method*

A.2 RReliefF

Regression ReliefF is an adaptive version of ReliefF for regression task. Relief's estimate ***W[A]*** of the quality of attribute *A* is an approximation of the following difference of probabilities where ***A*** is a vector of attributes A_i, $i = 1,\ldots,a$, where a is the number of explanatory attributes, and are labelled with the target value τ_j.

$$W[A] = P(\text{diff. value of } A|\text{nearest inst. from diff. class}) - P(\text{diff. value of } A|\text{nearest inst. from same class}) \quad (21)$$

In regression problems the predicted value $\tau(\cdot)$ is continuous, therefore (nearest) hits and misses cannot be used. To solve this difficulty, instead of requiring the exact knowledge of whether two instances belong to the same class or not, a kind of probability that the predicted values of two instances are different is introduced. This probability can be modelled with the relative distance between the predicted (class) values of two instances. To estimate ***W[A],*** the equation is reformulated, so that it can be directly evaluated using the probability that predicted values of two instances are different.

$$W[A] = \frac{P_{diffC|diffA}P_{diffA}}{P_{diffC}} - \frac{(1 - P_{diffC|diffA})P_{diffA}}{(1 - P_{diffC})} \quad (22)$$

where

P_{diffA} = *P(different value of A | nearest instances)*

P_{diffC} = *P(different prediction | nearest instances)*

and

$P_{diffC|diffA}$ = *P(diff. prediction | diff. value of A and nearest instances)*

Then, the term ***d(i, j)*** takes into account the distance between the two instances R_i and I_j. Rationale is that closer instances should have greater influence, so we exponentially decrease the influence of the instance I_j with the distance from the given instance R_i:

$$d(i,j) = \frac{d_1(i,j)}{\sum_{l=1}^{k} d_1(i,l)} \quad (23)$$

And

$$d_1(i,j) = e^{-\left(\frac{rank(R_i,I_j)}{\sigma}\right)^2} \quad (24)$$

where ***rank(R$_i$, I$_j$)*** is the rank of the instance ***I$_j$*** in a sequence of instances ordered by the distance from ***R$_i$*** and ***σ*** is a user defined parameter controlling the influence of the distance. Since we want to stick to the probabilistic interpretation of the results we normalize the contribution of each of ***k*** nearest instances by dividing it with the sum of all ***k*** contributions. The reason for using ranks instead of actual distances is that actual distances are problem dependent while by using ranks we assure that the nearest (and subsequent as well) instance always has the same impact on the weights.

A.3 MI

The Mutual Information based feature selection method we used in this study was from the extensively use Python library, scikit-learn. The purpose of this method is to measure the dependency between the variables in discrete and continuous features. It relies on nonparametric methods based on entropy estimation from ***k***-nearest neighbors' distances.

Consider a discrete variable ***X*** and the continuous variable ***Y***, drawn from probability density ***m(x,y).*** Both ***X*** and ***Y*** may be either univariate (composed of scalars) or multivariate (vectors). We will write discrete probability functions as ***p(·)*** and continuousdensities using the symbol $\mu(\cdot)$: therefore $p(x) = \int \mu(x,y)dy$ and $\mu(y) = \sum_x \mu(x,y)$. The mutual information is:

$$
\begin{aligned}
I(X,Y) &= H(X) + H(Y) - H(X,Y) \\
&= -\sum_x p(x)\log p(x) \\
&\qquad - \int \mu(y)\log\mu(y|x)dy \\
&\qquad + \sum_x \int \mu(x,y)\log\mu(x,y)dy \qquad (25) \\
&= -\int \mu(y)\log\mu(y)dy \\
&\qquad + \sum\nolimits_x \int \mu(x,y)\log\mu(y|x)dy \\
&= -\langle\log\mu(y)\rangle + \langle\log\mu(y|x)\rangle
\end{aligned}
$$

Here ***H*** denotes an entropy, ***μ(y)*** is the probability density for sampling *y* irrespective of the value of ***x***, and ***μ(y|x) = μ(x,y)/p(x)*** is the probability density for sampling ***y*** given a particular value of ***x***. The averages are taken over the full distribution and weighted by ***μ(x,y)***, and they would be straightforward to calculate if we knew the underlying density functions. Alternatively, each average can be taken over a representative set from ***(x,y)*** pairs sampled from the distribution; using this latter interpretation we estimate the MI from the mean of $\log\mu(y)$ and $\log\mu(y|x)$ at each of our sampled datapoints. The more points we have, the greater the accuracy. Finally, the *k*-nearest neighbor estimator was applied to estimate ***μ(y)*** by finding a neighbor from the full set of data points, and ***μ(y|x)*** by finding a neighbor in the subset of data points ***j*** for which $x_j = x_i$. The result of this method is the list of features ranked by its relative score to the target variable.

A.4 XGB

XGB is one of the most popular and efficient implementations of the Gradient Boosted Trees algorithm, a supervised learning method that is based on function

approximation by optimizing specific loss functions as well as applying several regularization techniques. Given a dataset $\boldsymbol{X} \in R^{N \times J}$ with N instances and J features, XGBoost predicts the $\boldsymbol{i}$-th instance $\boldsymbol{x}_i \in \mathrm{R}^{1 \times J}$ by using T regression functions

$$\hat{y}_i = \sum_{t=1}^{T} f_t(x_i) \tag{26}$$

Formally, let $\hat{\boldsymbol{y}}_{\boldsymbol{i}}^{(\boldsymbol{t})}$ be the prediction of the *i*-th instance at the *t*-th iteration, XGBoost is trained in an additive manner by adding $\boldsymbol{f_t}$ to minimize the following objective.

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) \tag{27}$$

Then, the original objective function is transformed to a function in the Euclidean domain, in order to be able to use traditional optimization techniques. A second-order Taylor expansion is used to approximate the loss function at the $\boldsymbol{t}$-th iteration as follows:

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n} \left[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{28}$$

Here, $\boldsymbol{l}(\cdot)$ is a differentiable convex loss function. For example, MSE loss is used for regression tasks and log-loss is for classification tasks. $\Omega(f_t) \triangleq \gamma + \frac{1}{2}\lambda \|w\|^2$ is the regularization function, where U is the number of leaves in the tree, γ and λ are parameters used to suppress tree number and weights

respectively. $g_i = \partial l\left(y_i, \hat{y}_i^{(t-1)}\right)$ and $h_i = \partial^2 l\left(y_i, \hat{y}_i^{(t-1)}\right)$ are the first- and second-order gradient statistics of the loss function at $\hat{y}_i^{(t-1)}$

Normally enumeration of all the possible tree structures is impossible. Instead the model starts from a single leaf node containing all instances. Then the node recursively splits the current instance set I to the left and right subset, denoted as I_L and I_R respectively. The loss reduction after the split is given by

$$\mathcal{L}_{split} \triangleq \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{\left(\sum_{i\in I} g_i\right)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma \qquad \textbf{(29)}$$

This formula is usually used in practice for evaluating the split candidates. When the stop condition (no positive loss reductions or max depth reached) is met, each leaf u can calculate its weight w according to the following equation

$$w = -\frac{\sum_{i\in I_u} g_i}{\sum_{i\in I_u} h_i + \lambda} \qquad \textbf{(30)}$$

A.5 CatBoost

CatBoost is a new gradient boosting algorithm that successfully works with categorical features with the lowest information loss. CatBoost differs from other gradient boosting algorithms. It is useful on small dataset and capable to handle category features. Namely, assume we observe a dataset of examples $D = \{(x_k, y_k)\}_{k=1,\ldots,n}$, where $x_k = (x_k^1, \ldots, x_k^m)$ is a random vector of *m* features and $y_k \in R$ is a target, which can be either binary or numerical. CatBoost uses an ordered target statistic to reduces overfitting and avoids target leakage. It performs a random permutation of the dataset and for each example we compute average label value for the example with the same category value placed before the

given one in the permutation. Their general idea is to compute the target statistics for x_k on a subset of examples $D_k \subset D \backslash \{x_k\}$ excluding x_k:

$$\hat{x}_k^i = \frac{\sum_{x_j \in D_k} 1_{x_k^i = x_k^j} \cdot y_j + ap}{\sum_{x_j \in D_k} 1_{x_k^i = x_k^j} + a} \quad (31)$$

where $a > 0$ is a parameter. A common setting for p is the average target value in the dataset.

This technique also ensures the use all the available past for each example to compute its target statistics and thereby encoding the categorical variables. Finally, CatBoost introduces ordered boosting to avoid prediction shift problem. In ordered boosting, a random permutation of the training examples is performed, and t different supporting models maintained (i-th model trained using only the first k samples in the permutation) and at each step residual or error is obtained by using previous model residuals.

REFERENCES

[1] H. Luo, S. Zhao, and R. Yao, “Determinants of Housing Prices in Dalian City, China: Empirical Study Based on Hedonic Price Model,” *Journal of Urban Planning and Development,* vol. 147, no. 2, pp. 05021017, 2021/06/01, 2021.

[2] P. Zhang, W. Ma, and T. Zhang, "Application of Artificial Neural Network to Predict Real Estate Investment in Qingdao," pp. 213-219, 2012.

[3] Q. Truong, M. Nguyen, H. Dang, and B. Mei, “Housing Price Prediction via Improved Machine Learning Techniques,” *Procedia Computer Science,* vol. 174, pp. 433-442, 2020/01/01/, 2020.

[4] J. Núñez-Tabales, J. Caridad, and F. Rey, “Artificial neural networks for predicting real estate prices,” *Revista de Metodos Cuantitativos para la Economia y la Empresa,* vol. 15, pp. 29-44, 06/01, 2013.

[5] N. Peter, H. Okagbue, O. Emmanuela C.M, and A. Akinola, “Review on the Application of Artificial Neural Networks in Real Estate Valuation,” *International Journal of Advanced Trends in Computer Science and Engineering,* vol. 9, pp. 2918 – 2925, 07/03, 2020.

[6] Y. E. Hamzaoui, and J. A. H. Perez, "Application of Artificial Neural Networks to Predict the Selling Price in the Real Estate Valuation Process." pp. 175-181.

[7] A. Mimis, A. Rovolis, and M. Stamou, “Property valuation with artificial neural network: The case of Athens,” *Journal of Property Research,* vol. 30, 06/01, 2013.

[8] P. Y. Wang, C. T. Chen, J. W. Su, T. Y. Wang, and S. H. Huang, “Deep Learning Model for House Price Prediction Using Heterogeneous Data Analysis Along With Joint Self-Attention Mechanism,” *IEEE Access,* vol. 9, pp. 55244-55259, 2021.

[9] C. Zhou, “House price prediction using polynomial regression with Particle Swarm Optimization,” *Journal of Physics: Conference Series,* vol. 1802, no. 3, pp. 032034, 2021/03/01, 2021.

[10] B. C. Csáji, “Approximation with artificial neural networks,” *Faculty of Sciences, Etvs Lornd University, Hungary,* vol. 24, no. 48, pp. 7, 2001.

[11] R. E. Bellman, *Adaptive control processes: a guided tour*: Princeton university press, 2015.

[12] R. Wang, "AdaBoost for Feature Selection, Classification and Its Relation with SVM, A Review," *Physics Procedia,* vol. 25, pp. 800-807, 2012/01/01/, 2012.

[13] S. Maozhun, and L. Ji, "Improved Garson algorithm based on neural network model." pp. 4307-4312.

[14] M. Robnik-Sikonja, and I. Kononenko, "An adaptation of Relief for attribute estimation in regression," *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, 02/11, 2000.

[15] M. Robnik-Šikonja, and I. Kononenko, "Theoretical and Empirical Analysis of ReliefF and RReliefF," *Machine Learning,* vol. 53, no. 1, pp. 23-69, 2003/10/01, 2003.

[16] M. Haindl, P. Somol, D. Ververidis, and C. Kotropoulos, "Feature Selection Based on Mutual Correlation." pp. 569-577.

[17] M. Hall, "Correlation-Based Feature Selection for Machine Learning," *Department of Computer Science,* vol. 19, 06/17, 2000.

[18] T. M. Cover, and J. A. Thomas, "Elements of information theory second edition solutions to problems," *Internet Access*, 2006.

[19] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal,* vol. 27, no. 3, pp. 379-423, 1948.

[20] G. U. Yule, *Biometrika,* vol. 36, no. 1/2, pp. 236-238, 1949.

[21] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine learning,* vol. 46, no. 1, pp. 389-422, 2002.

[22] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological),* vol. 58, no. 1, pp. 267-288, 1996.

[23] G. C. McDonald, and D. I. Galarneau, "A Monte Carlo Evaluation of Some Ridge-Type Estimators," *Journal of the American Statistical Association,* vol. 70, no. 350, pp. 407-416, 1975.

[24] D. Guan, W. Yuan, Y.-K. Lee, K. Najeebullah, and M. K. Rasel, "A review of

ensemble learning based feature selection," *IETE Technical Review,* vol. 31, no. 3, pp. 190-198, 2014.

[25] H. Jeon, and S. Oh, "Hybrid-Recursive Feature Elimination for Efficient Feature Selection," *Applied Sciences,* vol. 10, no. 9, 2020.

[26] B. Pes, "Ensemble feature selection for high-dimensional data: a stability analysis across multiple domains," *Neural Computing and Applications,* vol. 32, no. 10, pp. 5951-5973, 2020/05/01, 2020.

[27] B. Seijo-Pardo, I. Porto-Díaz, V. Bolón-Canedo, and A. Alonso-Betanzos, "Ensemble feature selection: Homogeneous and heterogeneous approaches," *Knowledge-Based Systems,* vol. 118, pp. 124-139, 2017/02/15/, 2017.

[28] S. Alelyani, "Stable bagging feature selection on medical data," *Journal of Big Data,* vol. 8, no. 1, pp. 11, 2021/01/07, 2021.

[29] N. S. Banu, and S. Suganya, "ENSEMBLE FEATURE SELECTION (EFS) AND ENSEMBLE HYBRID CLASSIFIERS (EHCS) FOR DIAGNOSIS OF SEIZURE USING EEG SIGNALS."

[30] S. S. Kshatri, D. Singh, B. Narain, S. Bhatia, M. T. Quasim, and G. R. Sinha, "An Empirical Analysis of Machine Learning Algorithms for Crime Prediction Using Stacked Generalization: An Ensemble Approach," *IEEE Access,* vol. 9, pp. 67488-67500, 2021.

[31] S. Rose, S. Nickolas, and S. Sangeetha, "A recursive ensemble-based feature selection for multi-output models to discover patterns among the soil nutrients," *Chemometrics and Intelligent Laboratory Systems,* vol. 208, pp. 104221, 2021/01/15/, 2021.

[32] I. Guyon, and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research,* vol. 3, no. Mar, pp. 1157-1182, 2003.

[33] M. Verleysen, and D. François, "The Curse of Dimensionality in Data Mining and Time Series Prediction." pp. 758-770.

[34] V. Bachu, and J. Anuradha, "A Review of Feature Selection and Its Methods," *Cybernetics and Information Technologies,* vol. 19, pp. 3, 03/19, 2019.

[35] A. L. Blum, and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence,* vol. 97, no. 1, pp. 245-271,

1997/12/01/, 1997.

[36] M. Kordos, “Data Selection for Neural Networks,” *Schedae Informaticae,* vol. 25, 03/24, 2017.

[37] J. Miao, and L. Niu, “A Survey on Feature Selection,” *Procedia Computer Science,* vol. 91, pp. 919-926, 2016/01/01/, 2016.

[38] B. Darst, K. Malecki, and C. Engelman, “Using recursive feature elimination in random forest to account for correlated variables in high dimensional data,” *BMC Genetics,* vol. 19, 09/17, 2018.

[39] P. Misra, and A. Singh, “Improving the Classification Accuracy using Recursive Feature Elimination with Cross-Validation,” vol. 11, pp. 659-665, 05/08, 2020.

[40] F. Song, Z. Guo, and D. Mei, "Feature Selection Using Principal Component Analysis." pp. 27-30.

[41] G. Doquire, and M. Verleysen, "Feature Selection with Mutual Information for Uncertain Data." pp. 330-341.

[42] P. Hanchuan, L. Fuhui, and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, no. 8, pp. 1226-1238, 2005.

[43] H. Zhou, Y. Zhang, Y. Zhang, and H. Liu, “Feature selection based on conditional mutual information: minimum conditional relevance and minimum conditional redundancy,” *Applied Intelligence,* vol. 49, no. 3, pp. 883-896, 2019/03/01, 2019.

[44] B. Frénay, G. Doquire, and M. Verleysen, “Is mutual information adequate for feature selection in regression?,” *Neural Networks,* vol. 48, pp. 1-7, 2013/12/01/, 2013.

[45] P. Carmona, J. Sotoca, F. Pla, F. K. H. Phoa, and J. Bioucas-Dias, *Feature Selection in Regression Tasks Using Conditional Mutual Information*, 2011.

[46] B. C. Ross, “Mutual Information between Discrete and Continuous Data Sets,” *PLOS ONE,* vol. 9, no. 2, pp. e87357, 2014.

[47] J. Martínez Sotoca, and F. Pla, “Supervised feature selection by clustering using conditional mutual information-based distances,” *Pattern Recognition,* vol. 43,

no. 6, pp. 2068-2081, 2010/06/01/, 2010.

[48] M. Beraha, A. M. Metelli, M. Papini, A. Tirinzoni, and M. Restelli, *Feature Selection via Mutual Information: New Theoretical Insights*, 2019.

[49] D. Loann, "A Review on Variable Selection in Regression Analysis," *Econometrics,* vol. 6, pp. 45, 11/23, 2018.

[50] K. Kira, and L. A. Rendell, "A Practical Approach to Feature Selection," *Machine Learning Proceedings 1992*, D. Sleeman and P. Edwards, eds., pp. 249-256, San Francisco (CA): Morgan Kaufmann, 1992.

[51] J. Golay, M. Leuenberger, and M. Kanevski, "Feature selection for regression problems based on the Morisita estimator of intrinsic dimension," *Pattern Recognition,* vol. 70, pp. 126-138, 2017/10/01/, 2017.

[52] J. Golay, and M. Kanevski, "A New Estimator of Intrinsic Dimension Based on the Multipoint Morisita Index," *Pattern Recognition,* vol. 48, pp. 4070-4081, 12/01, 2015.

[53] Y. Huang, and G. Hewings, "More Reliable Land Price Index: Is There a Slope Effect?," *Land,* vol. 10, no. 3, 2021.

[54] Y. Yu, J. Lu, D. Shen, and B. Chen, "Research on real estate pricing methods based on data mining and machine learning," *Neural Computing and Applications,* vol. 33, pp. 1-13, 05/01, 2021.

[55] W. Mingbo, T. Pei, W. Wang, S. Guo, C. Song, J. Chen, and C. Zhou, "Roles of locational factors in the rise and fall of restaurants: A case study of Beijing with POI data," *Cities,* vol. 113, pp. 103185, 06/01, 2021.

[56] B. Y. An, R. W. Bostic, A. Jakabovics, A. W. Orlando, and S. Rodnyansky, "Why Are Small and Medium Multifamily Properties So Inexpensive?," *The Journal of Real Estate Finance and Economics,* vol. 62, no. 3, pp. 402-422, 2021/04/01, 2021.

[57] K. Dong, C.-T. Chang, S. Wang, and X. Liu, "The Dynamic Correlation among Financial Leverage, House Price, and Consumer Expenditure in China," *Sustainability,* vol. 13, no. 5, 2021.

[58] W.-S. Lin, J.-C. Tou, S.-Y. Lin, and M.-Y. Yeh, "Effects of socioeconomic factors on regional housing prices in the USA," *International Journal of Housing Markets and Analysis,* vol. 7, 02/25, 2014.

[59] Y. Gu, “What are the most important factors that influence the changes in London Real Estate Prices? How to quantify them?,” *arXiv: Applications*, 2018.

[60] F. Riccioli, R. Fratini, and F. Boncinelli, “The Impacts in Real Estate of Landscape Values: Evidence from Tuscany (Italy),” *Sustainability,* vol. 13, no. 4, 2021.

[61] O. Aluko, “The effects of location and neighbourhood attributes on housing values in metropolitan Lagos,” *Journal of Geography and Regional Planning,* vol. 4, pp. 767-775, 2011.

[62] U. Musa, and W. Yusoff, "Impact of Location and Dwelling Characteristics on Residential Property Prices/Values: A Critical Review of Literature."

[63] L. Fernandez-Duran, A. Llorca, N. Ruiz, S. Valero, and V. Botti, “The impact of location on housing prices: applying the Artificial Neural Network Model as an analytical tool,” *ERSA conference papers*, 01/01, 2011.

[64] E. Boucq, and F. Papon, “Assessment of the Real Estate Benefits Due to Accessibility Gains Brought by a Transport Project: The Impacts of a Light Rail Infrastructure Improvement in the Hauts-de-Seine Department,” *European Transport \ Trasporti Europei,* vol. 40, pp. 51-68, 01/01, 2008.

[65] R. Cordera, P. Coppola, L. dell'Olio, and Á. Ibeas, “The impact of accessibility by public transport on real estate values: A comparison between the cities of Rome and Santander,” *Transportation Research Part A: Policy and Practice,* vol. 125, pp. 308-319, 2019/07/01/, 2019.

[66] J. D. Olden, M. K. Joy, and R. G. Death, “An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data,” *Ecological Modelling,* vol. 178, no. 3, pp. 389-397, 2004/11/01/, 2004.

[67] J. de Oña, and C. Garrido, “Extracting the contribution of independent variables in neural network models: a new approach to handle instability,” *Neural Computing and Applications,* vol. 25, no. 3, pp. 859-869, 2014/09/01, 2014.

[68] Y. Freund, and R. E. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” *Journal of Computer and System Sciences,* vol. 55, no. 1, pp. 119-139, 1997/08/01/, 1997.

[69] D. Shrestha, and D. Solomatine, “Experiments with AdaBoost.RT, an Improved Boosting Scheme for Regression,” *Neural Computation,* vol. 18, pp. 1678-1710,

07/01, 2006.

[70] H. Drucker, “Improving Regressors Using Boosting Techniques,” *Proceedings of the 14th International Conference on Machine Learning*, 08/17, 1997.

[71] J. A. Kahn, “What Drives Housing Prices?,” *Economic Growth*, 2008.

[72] S. Kim, and H. Kim, “A new metric of absolute percentage error for intermittent demand forecasts,” *International Journal of Forecasting,* vol. 32, no. 3, pp. 669-679, 2016/07/01/, 2016.

[73] K. Bache, and M. Lichman, "UCI Machine Learning Repository," 2013.

VITA

NAME	Kankawee Chanasit
DATE OF BIRTH	19 August 1995
PLACE OF BIRTH	Chanthaburi
INSTITUTIONS ATTENDED	Faculty of Engineering, Chulalongkorn University.
HOME ADDRESS	403 Bamboo for rest apartment 90 Sukhumvit 52 Prakhanong Klongtoey Bangkok 10260
PUBLICATION	P. Wiriyachaiporn, K. Chanasit, A. Suchato, P. Punyabukkana and E. Chuangsuwanich, "Algorithmic Music Composition Comparison," 2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2018, pp. 1-6, doi: 10.1109/JCSSE.2018.8457397.
	K. Chanasit, E. Chuangsuwanich, A. Suchato and P. Punyabukkana, "A Real Estate Valuation Model Using Boosted Feature Selection," in IEEE Access, vol. 9, pp. 86938-86953, 2021, doi: 10.1109/ACCESS.2021.3089198.