# Encoding Techniques for Handling Categorical Data in Machine Learning-Based Software Development Effort Estimation

Mohamed Hosni[a]
*MOSI Research Team, ENSAM, University Moulay Ismail of Meknes, Meknes, Morocco*

Keywords: Categorical Data, Encoder, Software Effort Estimation, Ensemble Effort Estimation.

Abstract: Planning, controlling, and monitoring a software project primarily rely on the estimates of the software development effort. These estimates are usually conducted during the early stages of the software life cycle. At this phase, the available information about the software product is categorical in nature, and only a few numerical data points are available. Therefore, building an accurate effort estimator begins with determining how to process the categorical data that characterizes the software project. This paper aims to shed light on the ways in which categorical data can be treated in software development effort estimation (SDEE) datasets through encoding techniques. Four encoders were used in this study, including one-hot encoder, label encoder, count encoder, and target encoder. Four well-known machine learning (ML) estimators and a homogeneous ensemble were utilized. The empirical analysis was conducted using four datasets. The datasets generated by means of the one-hot encoder appeared to be suitable for the ML estimators, as they resulted in more accurate estimation. The ensemble, which combined four variants of the same technique trained using different datasets generated by means of encoder techniques, demonstrated an equal or better performance compared to the single ML estimation technique. The overall results are promising and pave the way for a new approach to handling categorical data in SDEE datasets.

## 1 INTRODUCTION

Software development effort estimation (SDEE) is defined as the process of determining the amount of effort required to develop a software system (Wen et al., 2012). Delivering an accurate estimate is crucial for the success of the software development project, as an inaccurate estimate can lead to project failure (Oliveira et al., 2010). To avoid such problems, over the past five decades, researchers have developed and evaluated several estimation techniques aimed at providing an accurate estimate of the effort required to develop a software project (Ali and Gravino, 2019). These estimation techniques can be classified into three main categories (Jorgensen and Shepperd, 2006): expert judgment, algorithmic techniques, and machine learning (ML) techniques.

Wen et al. conducted a systematic literature review (SLR) on the use of machine learning (ML) techniques in SDEE studies conducted between 1991 and 2010 (Wen et al., 2012). They identified eight different ML techniques that had been investigated in the 84 selected studies. Among these techniques, Analogy and Artificial Neural Networks (ANN) were the most frequently adopted. The review highlighted the increasing popularity of ML techniques in SDEE over the past three decades and demonstrated that estimates produced using these techniques were more accurate compared to other estimation methods. In recent literature, a new approach called Ensemble Effort Estimation (EEE) has been proposed, which involves combining multiple SDEE techniques under a specific combination rule (Kocaguneli et al., 2011). The overall conclusions drawn from the literature of SDEE indicate that this new approach generates more accurate estimates compared to using a single technique (Cabral et al., 2023).

Nevertheless, accurately estimating the effort required for software development is a complex undertaking, especially in the early stages of the software life cycle when the available information tends to be more descriptive rather than quantifiable (Amazal and Idri, 2019). Indeed, predictive techniques rely on constructing their models using a set of features (i.e., cost drivers) that define software projects. Most SDEE techniques generate predictions by using numerical

[a] https://orcid.org/0000-0001-7336-4276

Hosni, M.
Encoding Techniques for Handling Categorical Data in Machine Learning-Based Software Development Effort Estimation.
DOI: 10.5220/0012259400003598
In *Proceedings of the 15th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2023) - Volume 1: KDIR*, pages 460-467
ISBN: 978-989-758-671-2; ISSN: 2184-3228

attributes. However, during the initial stages of the software life cycle, the available information tends to be more categorical than numerical. Furthermore, the historical datasets contain a significant number of categorical features, such as COCOMO 81 and ISBSG datasets, among others.

The categorical attributes within SDEE can be measured either nominally or ordinally. To address this limitation, various approaches have been explored in the existing literature (Angelis et al., 2001; Li et al., 2007). These techniques include the use of Euclidean and Manhattan distance, decision tree-based classification, fuzzy logic, and the grey relational coefficient. Many of the proposed methods for handling categorical attributes employ hybrid techniques, combining a specific approach for categorical attribute handling with the SDEE technique itself. As a result, the process of constructing the estimation technique and the preprocessing stage become inseparable. For instance, distance measures like Euclidean and Manhattan distance are specific to predictive techniques that predict effort based on similarity, making them specific to this type of predictive technique and not applicable to other SDEE techniques such as Decision Trees (DT), Support Vector Regression (SVR), or ANN.

Indeed, handling categorical attributes is considered a part of the feature engineering process, which is a key step in data preprocessing. It involves converting categorical attributes into their numerical counterparts. It's important to note that preparing the data for a predictive model is a distinct task from the modeling process. Once the original data has been preprocessed and all the categorical attributes have been transformed into numerical format, the modeling process (i.e., building the SDEE technique) can begin independently from the preprocessing stage.

This paper aims to examine the utilization of categorical preprocessing techniques and evaluate their impact on the predictive performance of SDEE techniques, including both single and ensemble methods. The original dataset, which includes categorical features, is preprocessed using different encoder techniques to convert them into numerical attributes. Four encoder techniques, namely One-Hot encoding, Label encoding, Counter encoding, and Target encoding, are used for that purpose. Next, several ML techniques are constructed, including K-nearest neighbor, DT, SVR, and Multilayer Perceptron (MLP) neural networks. These techniques are trained using datasets generated from the original dataset, with each dataset being processed using a different encoder technique. Furthermore, an ensemble is created by combining four variants of the same technique, each trained on a dataset generated using a different encoder technique. The final output of ensemble is generated by the average rule. The predictive capabilities of these methods are assessed using various unbiased metrics.

Toward these aims, two research questions (RQs) have been examined:

- **(RQ1):** Among the four encoder techniques employed in this study, which one yields superior performance when used with a single ML technique?
- **(RQ2):** Does the utilization of different encoder techniques alongside the same SDEE technique (ensemble technique) result in more accurate estimates compared to using a single SDEE technique?

The primary contributions of this empirical work are as follows:

- The application of four encoder techniques to process categorical attributes in SDEE datasets.
- Investigation of the impact of encoders on the accuracy of SDEE techniques.
- Assessment of the effect of encoders on the performance of the EEE approach.

To the best of our knowledge, no existing research has explored the impact of encoder techniques on the predictive capabilities of SDEE-based ML techniques.

The remainder of this paper is structured as follows: Section 2 presents the main findings of related work in literature. Section 3 defines the feature encoding techniques utilized in this study. Section 4 provides the empirical design employed in this research. Section 5 discusses the empirical results obtained from the experiments. Section 6 presents future work and concludes the paper.

## 2 RELATED WORK

Wen et al. conducted a SLR to investigate the utilization of SDEE based on ML techniques for estimating software development effort (Wen et al., 2012). The review examined 84 papers published between 1991 and 2010. Through this review, eight ML techniques were identified: CBR or analogy, ANN, DTs, bayesian networks, SVR, genetic algorithms, genetic programming, and association rules. These techniques have been employed to estimate software development effort. Among them, CBR and ANN were found to be the most commonly used techniques, accounting for 37 and 26 respectively. The review also revealed that the estimation derived from ML techniques, particularly CBR and ANN, exhibited higher accuracy compared to non-ML techniques. Another recent review conducted by Asad et al. focused on the application of ML techniques in SDEE (Ali and Gravino, 2019). This review highlighted that ANN and SVR techniques were extensively investigated in the literature. Furthermore, both techniques were found to generate more accurate estimates compared to other ML and non-ML techniques. In terms of performance metrics, both reviews identified Mean Magnitude of Relative Error (MMRE) and Prediction within 25 of actual effort (Pred(25)) as commonly used performance measures.

Recently, there has been growing interest in exploring ensemble methods in the context of SDEE (Hosni et al., 2018a; Azhar et al., 2013). This approach involves predicting software development effort by combining multiple techniques using a specific combination rule. Two types of ensembles are defined: homogeneous ensembles, which combine different variants of the same estimation technique, and heterogeneous ensembles, which incorporate multiple different estimation techniques. Idri et al. conducted a SLR to examine the use of EEE (Idri et al., 2016). Their review analyzed 24 papers published between 2000 and 2016. The findings revealed that 16 SDEE techniques were employed to construct EEE, with the homogeneous ensemble being the most commonly used type. Additionally, 20 combiners were utilized to combine the output of single estimators, and linear rules were the most frequently adopted. In terms of performance accuracy, the review indicated that the ensemble approach consistently generated more accurate results than using a single estimator. A recent update of this SLR was conducted by Cabral et al. (Cabral et al., 2023), and their findings aligned with those reported by Idri et al. They also observed an increase in research work in the last five years, driven by the promising results obtained through the ensemble approach.

The treatment of categorical data in SDEE datasets has been the focus of several papers in the literature. Amazal et al. conducted a systematic mapping study to explore the handling of categorical data in SDEE (Amazal and Idri, 2019). The review included 27 papers that addressed this topic. The findings revealed that Euclidean distance, fuzzy logic, and fuzzy clustering techniques were commonly employed to handle categorical data, particularly when utilizing the analogy technique. On the other hand, when using regression techniques, most papers utilized Analysis of Variance and combinations of categories. Furthermore, the review identified several SDEE techniques that have been investigated in the literature, namely analogy, Regression, and Classification and Regression Trees (CART). Additionally, various techniques were used to handle categorical features, including Euclidean distance, classification by DT, fuzzy clustering, grey relational coefficient, and local similarity. These techniques were combined with specific SDEE techniques, resulting in hybrid methods. The objective of these approaches, as explored in the literature, is to enhance the accuracy of specific SDEE techniques when working with datasets that contain categorical effort drivers.

However, handling categorical attributes is typically considered as part of the feature engineering phase in the ML process. This phase is essential and occurs before constructing an ML model. The focus of this paper is to address categorical attributes independently of the predictive model employed. In other words, the aim is to generate a dataset comprising just numerical attributes obtained through a specific encoder technique applied to the original dataset that contains categorical attributes (Breskuvienė and Dzemyda, 2023), (De La Bourdonnaye and Daniel, 2021). By doing so, the paper aims to preprocess the data and transform the categorical attributes into numerical representations, which can then be further used as input for different ML estimators.

## 3 FEATURE ENCODING TECHNIQUES

This section provides a brief description of the four features encoding techniques used in this paper. These techniques differ from each other in the process of transforming the categorical attributes into numerical ones. In fact, the process of transforming categorical attributes into numerical ones is crucial for ML technique to effectively process (Breskuvienė and Dzemyda, 2023; De La Bourdonnaye and Daniel, 2021).

**One Hot Encoder:** is a technique used to convert a categorical attribute into a numerical representation. The process begins by determining the number of distinct categories within the categorical attribute. Subsequently, a set of columns is created, with each column representing one of the distinct categories. Binary values (0 or 1) are then assigned to these columns based on whether a data point belongs to a particular category or not. The number of columns generated is equal to the number of distinct categories present in the attribute.

**Label Encoder:** is a simple encoding technique. It consists of assigning a unique numerical value of each category value presents in the categorical attribute. The numerical value starts by 0 or 1 and it increments

for each category. This technique can work for both types of categorical attributes ordinal and nominal.
**Count Encoder:** also known as counting encoder, is a technique that replaces categorical values with numerical values based on the number of occurrences of each category in an attribute vector. This encoder provides additional information to the dataset by considering the frequency of a specific category's appearance, rather than solely relying on the categorical value of the attribute.
**Target Encoder:** also known as target-based encoding, the idea of this encoder involves replacing the categorical values of a categorical feature with numerical values that reflect the relationship between each category and the target variable. Instead of using the original categorical value, a statistical value derived from the target variable is used. This statistical value can be the median, average, or mode value of the target attribute, for instances belonging to that specific category. In this study, the average value of the target variable was adopted as the statistical value for target encoding.

## 4 EMPIRICAL DESIGN

### 4.1 Machine Learning Used

Four machine learning (ML) techniques, namely KNN, SVR, MLP and DT, have been selected to build the SDEE techniques. These techniques are commonly used in SDEE literature (Kocaguneli et al., 2011; Kocaguneli et al., 2009; Hosni et al., 2018b). Additionally, a homogeneous ensemble has been created using the average combiner. In this ensemble, four variants of a specific ML technique are trained on the four preprocessed datasets. Each variant is trained independently in a preprocessed dataset by means of a specific encoder technique, and their predictions are combined using the average combiner.

### 4.2 Performance Metrics and Statistical Test

Most of SLRs conducted in the literature of SDEE claim that the MMRE and Prediction level are the most frequently used to assess the accuracy performance SDEE techniques. These performance criteria are based on the mean relative error (MRE). This criterion was criticized by several researchers in literature for being biased towards underestimation. To avoid this shortcoming several alternative performance metrics have been proposed such as Mean Absolute Error (MAE), Mean Balanced Relative Error (MBRE), Mean Inverted Balanced Relative Error (MIBRE) which are considered less venerable to bias and asymmetry, and Logarithmic Standard Deviation (LSD) (Miyazaki et al., 1991; Foss et al., 2003). Therefore, in this paper we adopted these metrics along with their median value, except for LSD and Pred(25). Another criterion was proposed by Shepperd and MacDonell called Standardized Accuracy (SA) (Kocaguneli and Menzies, 2013). This criterion compares a predictive model to a baseline estimator created by a random guessing approach.

To verify if the predictions of a model are generated by chance and if there is an improvement over random guessing the Effect Size criterion was used.

The leave-one-out cross validation (LOOCV) technique was used to construct our proposed predictive techniques.

The Scott-Knott (SK) statistical test based on AE was performed to identify the techniques that share similar predictive capabilities.

### 4.3 Datasets

Four datasets were chosen for the empirical analysis conducted in this paper. Three of these datasets were selected from the PROMISE repository, while one dataset was obtained from the ISBSG. These datasets were suitable for the experiments raised in this paper since they contain a large number of categorical attributes. Table 1 provides an overview of the characteristics of the five selected datasets.

Table 1: Characteristics of the employed datasets.

| Dataset | Size | Numerical | Categorical | Effort | | | |
|---|---|---|---|---|---|---|---|
| | | | | Min | Max | Mean | Median |
| ISBSG | 266 | 6 | 5 | 47 | 54620 | 4790.10 | 2322.50 |
| Nasa93 | 93 | 2 | 20 | 8.4 | 8211 | 624.41 | 252 |
| Maxwell | 63 | 2 | 22 | 583 | 63694 | 8109.54 | 5100 |
| USP05 | 203 | 1 | 13 | 0.5 | 400 | 11.58 | 3 |

### 4.4 Methodology Used

The following steps were followed for each dataset to build the proposed SDEE techniques in this study:
**For the Single SDEE Techniques:**
**Step 1:** Categorical feature transformation: The selected encoder techniques, namely One Hot Encoding (OH), Label Encoding (LE), Counting Encoding (CE), and Target Encoding (TE), were applied to transform categorical features into numerical representations. This resulted in four distinct datasets.
**Step 2:** ML technique construction: Each of the four ML techniques was built using the grid search optimization technique and 10-fold cross-validation.
**Step 3:** Parameter selection: The optimal parame-

ters for each ML technique were selected based on the specific dataset.
**Step 4:** Reasonability assessment: The optimized ML techniques were evaluated using performance metrics, such as SA and effect size, to determine their effectiveness.
**Step 5:** Performance evaluation: The selected ML techniques were assessed in terms of eight performance metrics MAE, MdAE, MIBRE, MdIBRE, MBRE, MdBRE, Pred(25) and LSD). The LOOCV technique was employed for this evaluation.
**Step 6:** Statistical analysis: Perform statistical analysis based on the AE using the SK test.
**For the Ensemble Methods:**
**Step 1:** Homogeneous ensemble construction: For each ML technique, build four variants technique in each of the four preprocessed datasets and combine their estimates using the average combiner.
**Step 2:** Ensemble performance evaluation: The performance accuracy of the constructed ensembles was measured using the performance metrics mentioned earlier.
**Step 3:** Techniques ranking: The constructed techniques, both single ML techniques and ensembles, were ranked using the Borda count voting system based on the performance metrics.
**Step 4:** Statistical analysis: Statistical analysis was performed using the SK test based on AE.
To ensure clarity, we utilized the following abbreviations: **SDEE technique + Encoder**. For instance, **KNNOH** represents the KNN technique trained on a dataset where categorical features were transformed into numerical ones using the One Hot (OH) encoding technique.

## 5 EMPIRICAL ANALYSIS

### 5.1 Single SDEE Evaluation

We initiate our empirical analysis by preprocessing the data to convert categorical attributes into numerical values. This procedure was performed on each dataset using the four encoders employed in this paper. Table 2 presents the feature counts for each dataset based on the encoder utilized. The results indicate that the number of features increased across all datasets when the one-hot encoder was applied. This outcome is expected since the one-hot encoder creates a column for each category within a categorical attribute, and some attributes contain more than eight categories. For the remaining encoders (label encoding, count encoder, and target encoder), the number of features in the processed dataset remains the same as the original dataset. This is because these encoders replace categories with numerical values based on either the category's frequency or its relationship with the target variable. A total of 16 datasets were utilized in this study, as each dataset was processed using the four encoders.

Table 2: Number of features after the encoding step.

| Encoder | Maxwell | ISBSG | Nasa93 | USP05 |
|---|---|---|---|---|
| **OH** | 93 | 58 | 85 | 302 |
| **LE** | 24 | 11 | 22 | 14 |
| **TE** | 24 | 11 | 22 | 14 |
| **CE** | 24 | 11 | 22 | 14 |

Next, we proceed to build our estimation techniques using the grid search optimization technique to determine the optimal parameter values that minimize the MAE in each of the 16 datasets. This process is carried out using the 10-fold cross-validation technique. To assess the performance of the constructed techniques, we compare them to a baseline estimator, specifically the random guessing estimator. The performance of the four ML techniques is compared against the 5% quantile of this baseline estimator. The results obtained suggest that the developed techniques consistently outperform the baseline method across all datasets. Furthermore, all the techniques exhibit significant improvements over the random guessing estimator. Among the techniques, the KNN technique achieves the highest level of accuracy in all datasets, regardless of the encoder technique used for preprocessing. On the other hand, the SVR technique falls short of achieving 75 accuracy in terms of SA in all datasets. The DT and MLP techniques generally achieve higher levels of accuracy, with the exception of the Maxwell dataset, where the MLPCE has the lowest SA value.

Afterwards, the performance accuracy of the proposed techniques is assessed using eight performance criteria through the LOOCV technique. The final ranking is determined using the Borda count voting system based on the selected accuracy criteria. The use of eight performance criteria is justified by the fact that different criteria can result in different rankings for a given estimator, as each criterion captures a specific aspect of performance accuracy. Table 3 displays the rankings obtained from the voting system. The rankings reveal that the DT and KNN techniques consistently dominate the top six positions across all datasets, regardless of the encoder used. The only exception is the Maxwell dataset, where the MLPOH technique is ranked fourth.

The key findings from the ranking are as follows:

- The KNN technique achieved the top rank in the

Table 3: Rank of the 16 SDEE techniques.

| Rank | Maxwell | ISBSG | Nasa93 | USP05 |
|---|---|---|---|---|
| 1 | DTOH | KNNLE | DTOH | KNNLE |
| 2 | KNNTE | KNNCE | DTLE | KNNOH |
| 3 | KNNOH | KNNOH | DTCE | KNNTE |
| 4 | MLPOH | KNNTE | KNNOH | DTOH |
| 5 | KNNCE | DTTE | DTTE | KNNCE |
| 6 | DTLE | DTCE | KNNCE | DTCE |
| 7 | MLPLE | MLPOH | MLPLE | DTTE |
| 8 | SVROH | DTOH | SVROH | SVROH |
| 9 | MLPTE | DTLE | KNNLE | MLPCE |
| 10 | DTCE | MLPTE | MLPCE | SVRTE |
| 11 | KNNLE | MLPLE | MLPTE | SVRCE |
| 12 | DTTE | MLPCE | KNNTE | SVRLE |
| 13 | SVRCE | SVROH | SVRTE | MLPOH |
| 14 | SVRTE | SVRTE | SVRLE | MLPTE |
| 15 | SVRLE | SVRLE | SVRCE | DTLE |
| 16 | MLPCE | SVRCE | MLPOH | MLPLE |

ISBSG and USP05 datasets when LE technique was used. In the remaining datasets, where the OH encoder was used, the DT technique obtained the highest rank.

- The SVR techniques consistently received lower rankings, with the ISBSG dataset showing the lowest positions.
- Three MLP techniques (trained in dataset preprocessed by CE, OH, LE techniques) were ranked last in three datasets.
- MLPOH achieved higher performance accuracy compared to other MLP techniques when using the remaining encoders in the Maxwell and ISBSG datasets. In the NASA93 dataset, MLPLE demonstrated greater accuracy than the other MLP techniques, while in the USP05 dataset, MLPCE outperformed the others.
- SVROH outperformed the remaining SVR techniques in all datasets.
- DTOH achieved a higher rank than other DT techniques in all datasets, except for the ISBSG dataset where DTTE was ranked higher.
- KNNLE outperformed other KNN techniques in two datasets, while KNNTE and KNNOH achieved better positions than other KNN techniques in one dataset each.

To further validate the obtained results, we performed the Scott-Knott test to cluster the different techniques with similar predictive properties based on their AE.

The SK test identified varying numbers of clusters in each dataset with different techniques. Specifically, it identified 11 clusters in the Maxwell dataset, eight clusters in the USP05 dataset, six clusters in the Nasa93 dataset, and four clusters in the ISBSG dataset.

The main observations than can be noticed are:

- In the Maxwell dataset, the best cluster consists of three different ML techniques: DTOH, KNNTE, and MLP with OH and LE encoders. On the other hand, the worst cluster contains the MLPCE technique.
- In the ISBSG dataset, the best cluster comprises the four KNN techniques, while the worst cluster contains three MLP techniques and the four SVR techniques.
- In the Nasa93 dataset, the best cluster includes six techniques: the four DT techniques and two KNN techniques (with OH and CE encoders). The last cluster contains the MLPOH and SVRCE techniques.
- In the USP05 dataset, the best cluster consists of six techniques: the four KNN techniques and two DT techniques (DTCE and DTOH). The MLPE technique forms the worst cluster. Based on these empirical findings, we can draw the following conclusions:
- The KNN technique emerges as the most accurate among the techniques used in this study, as one of its variants belongs to the best cluster identified by the SK test in all datasets.
- The DT technique produces more accurate estimates, as at least one of its variants belongs to the best cluster in three datasets.
- The SVR techniques perform poorly in this study, as none of their variants belong to the best cluster.
- The MLP technique appears to be a viable alternative to the DT and KNN techniques in the Maxwell dataset, as two of its variants (MLPOH and MLPLE) are members of the best cluster in this dataset.

Regarding the encoder techniques, Table 4 provides the frequency of each encoder in the best cluster identified by the SK test in each dataset. It is observed that the OH encoder appears seven times in the best cluster across all datasets, followed by the CE technique, which appears five times. The remaining two techniques appear four times. These results suggest a slight preference for the OH encoding technique compared to the other techniques.

## 5.2 Ensemble Methods Evaluation

The next step involves constructing a homogeneous ensemble that incorporates four variants of the same

Table 4: Number of occurrences of each encoder in the best cluster.

| Encoder | Maxwell | ISBSG | Nasa93 | USP05 | Total |
|---|---|---|---|---|---|
| **OH** | 2 | 1 | 2 | 2 | 7 |
| **CE** | 0 | 1 | 2 | 2 | 5 |
| **LE** | 1 | 1 | 1 | 1 | 4 |
| **TE** | 1 | 1 | 1 | 1 | 4 |

ML technique trained on the four processed datasets using the average rule. All the constructed ensembles outperform the 5% quantile of random guessing, as their members achieve the same, and the combination rule used is a linear rule. The EAKNN ensemble performs the best in the ISBSG and USP05 datasets, while the EADT ensemble is the top estimator in the remaining two datasets.The EASVR ensemble consistently ranks last in all datasets.

Table 5 displays the rankings obtained using the Borda count method for twenty techniques based on the performance indicators used. It is observed that none of the developed ensembles achieved a top-four ranking in all datasets, with the best position achieved by the EAKNN ensemble in the ISBSG dataset. However, none of the ensembles obtained the last position in any dataset. To further validate these results, the SK based on AE was performed.

Table 5: Rank of the twenty SDEE techniques, ensembles are in bold.

| Rank | Maxwell | ISBSG | Nasa93 | USP05 |
|---|---|---|---|---|
| **1** | DTOH | KNNLE | DTOH | KNNLE |
| **2** | KNNTE | KNNCE | DTLE | KNNOH |
| **3** | KNNOH | KNNOH | DTCE | KNNTE |
| **4** | MLPOH | KNNTE | KNNOH | DTOH |
| **5** | KNNCE | **EAKNN** | DTTE | KNNCE |
| **6** | DTLE | DTTE | KNNCE | **EAKNN** |
| **7** | MLPLE | DTCE | EADT | DTCE |
| **8** | **EAKNN** | **EADT** | **EAKNN** | DTTE |
| **9** | SVROH | MLPOH | MLPLE | **EADT** |
| **10** | **EADT** | **EAMLP** | **EAMLP** | SVROH |
| **11** | MLPTE | DTOH | SVROH | MLPCE |
| **12** | DTCE | DTLE | KNNLE | **EASVR** |
| **13** | **EAMLP** | MLPTE | **EASVR** | SVRTE |
| **14** | KNNLE | MLPLE | MLPCE | SVRCE |
| **15** | DTTE | MLPCE | MLPTE | SVRLE |
| **16** | **EASVR** | SVROH | KNNTE | MLPOH |
| **17** | SVRCE | SVRTE | SVRTE | **EAMLP** |
| **18** | SVRTE | **EASVR** | SVRLE | MLPTE |
| **19** | SVRLE | SVRLE | SVRCE | DTLE |
| **20** | MLPCE | SVRCE | MLPOH | MLPLE |

Different clusters have been identified in every dataset. The Maxwell dataset had the largest number of clusters with 11, followed by USP05 with nine clusters. Nasa93 had seven clusters, and ISBSG had five clusters. From the SK test results, the following observations can be made:

- In the Maxwell dataset, the best cluster does not include any ensemble technique. However, none of the ensembles belonged to the worst cluster.
- In the ISBSG dataset, the EKNN ensemble is part of the best cluster along with the four variants of the KNN technique. The EASVR ensemble is found in the worst cluster.
- In the Nasa93 dataset, the EADT ensemble is part of the best cluster along with its four constituents and two KNN variants. Additionally, the remaining three ensembles were not grouped in the worst cluster.
- In the USP05 dataset, the EAKNN ensemble is found in the best cluster along with its members and two DT variants.

In summary, none of the ensemble techniques appear to be consistently more accurate than all variants of the four ML techniques constructed in this study. However, the combination of multiple accurate techniques, such as EADT and EAKNN, generates more accurate predictions than other ML techniques. Moreover, except for the EASVR ensemble, none of the ensemble methods perform worse than all the ML estimators constructed in this study. Based on these results, we can conclude that the ensemble methods achieve equal or better performance than the individual techniques constructed, providing evidence for their effectiveness.

## 6 CONCLUSION AND FURTHER WORK

This study investigated the impact of encoder techniques on the prediction accuracy of SDEE techniques by processing categorical cost drivers in SDEE datasets. Four encoders were used to convert categorical attributes into numerical ones, resulting in four datasets. ML techniques (KNN, SVR, MLP, and DT) were optimized in each dataset using grid search optimization. A homogeneous ensemble was constructed by combining one estimation technique trained on each dataset using the average as a combiner. The proposed techniques were evaluated on the four datasets using various accuracy indicators through LOOCV. In short, the main findings related to the RQs discussed in this paper are as follows:
**(RQ1):** No conclusive evidence exists for the best encoder technique to enhance effort estimation performance. However, empirical results suggest that one hot encoding may be a favorable choice among the

encoders used in this study. Further experiments are required to reach a final conclusion on the best encoding technique.
**(RQ2):** The results show that both single techniques and the homogeneous ensemble developed in this study demonstrate similar predictive accuracy levels. In certain cases, the single KNN technique outperforms the ensemble technique, regardless of the encoder used for dataset processing.

Exploring alternative encoding techniques for processing categorical data in SDEE datasets is an important research direction. Additionally, investigating heterogeneous ensembles that incorporate different ML techniques trained on various processed datasets is crucial to determine whether encoder techniques can serve as a source of diversity in ensemble approaches.

## REFERENCES

Ali, A. and Gravino, C. (2019). A systematic literature review of software effort prediction using machine learning methods. *Journal of software: evolution and process*, 31(10):e2211.

Amazal, F. A. and Idri, A. (2019). Handling of categorical data in software development effort estimation: a systematic mapping study. In *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 763–770. IEEE.

Angelis, L., Stamelos, I., and Morisio, M. (2001). Building a software cost estimation model based on categorical data. In *Proceedings Seventh International Software Metrics Symposium*, pages 4–15. IEEE.

Azhar, D., Riddle, P., Mendes, E., Mittas, N., and Angelis, L. (2013). Using ensembles for web effort estimation. In *2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 173–182. IEEE.

Breskuvienė, D. and Dzemyda, G. (2023). Categorical feature encoding techniques for improved classifier performance when dealing with imbalanced data of fraudulent transactions. *INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL*, 18(3).

Cabral, J. T. H. d. A., Oliveira, A. L., and da Silva, F. Q. (2023). Ensemble effort estimation: An updated and extended systematic literature review. *Journal of Systems and Software*, 195:111542.

De La Bourdonnaye, F. and Daniel, F. (2021). Evaluating categorical encoding methods on a real credit card fraud detection database. *arXiv preprint arXiv:2112.12024*.

Foss, T., Stensrud, E., Kitchenham, B., and Myrtveit, I. (2003). A simulation study of the model evaluation criterion mmre. *IEEE Transactions on software engineering*, 29(11):985–995.

Hosni, M., Idri, A., and Abran, A. (2018a). Improved effort estimation of heterogeneous ensembles using filter feature selection. In *ICSOFT*, pages 439–446.

Hosni, M., Idri, A., Abran, A., and Nassif, A. B. (2018b). On the value of parameter tuning in heterogeneous ensembles effort estimation. *Soft Computing*, 22:5977–6010.

Idri, A., Hosni, M., and Abran, A. (2016). Systematic literature review of ensemble effort estimation. *Journal of Systems and Software*, 118:151–175.

Jorgensen, M. and Shepperd, M. (2006). A systematic review of software development cost estimation studies. *IEEE Transactions on software engineering*, 33(1):33–53.

Kocaguneli, E., Kultur, Y., and Bener, A. (2009). Combining multiple learners induced on multiple datasets for software effort prediction. In *International Symposium on Software Reliability Engineering (ISSRE)*.

Kocaguneli, E. and Menzies, T. (2013). Software effort models should be assessed via leave-one-out validation. *Journal of Systems and Software*, 86(7):1879–1890.

Kocaguneli, E., Menzies, T., and Keung, J. W. (2011). On the value of ensemble effort estimation. *IEEE Transactions on Software Engineering*, 38(6):1403–1416.

Li, J., Ruhe, G., Al-Emran, A., and Richter, M. M. (2007). A flexible method for software effort estimation by analogy. *Empirical Software Engineering*, 12:65–106.

Miyazaki, Y., Takanou, A., Nozaki, H., Nakagawa, N., and Okada, K. (1991). Method to estimate parameter values in software prediction models. *Information and Software Technology*, 33(3):239–243.

Oliveira, A. L., Braga, P. L., Lima, R. M., and Cornélio, M. L. (2010). Ga-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation. *information and Software Technology*, 52(11):1155–1166.

Wen, J., Li, S., Lin, Z., Hu, Y., and Huang, C. (2012). Systematic literature review of machine learning based software development effort estimation models. *Information and Software Technology*, 54(1):41–59.