ORIGINAL ARTICLE

Journal Section

# Unlocking the Potential of Weight of Evidence and Entity Embedding Encoding for Categorical Data Transformation in Medical Datasets: An Innovative Approach to Enhance Classification Accuracy

**Anitha M ME[1*] | Nickolas S PhD [1*] | Mary Saira bhanu S PhD[2†] | Gayathiri S ME[1‡]**

[1]Department of Computer Applications, National Institute of Technology, Tiruchirappalli, Tamil Nadu, 620015, India

[2]Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu, 620015, India

**Correspondence**
Department of Computer Applications, National Institute of Technology, Tiruchirappalli, Tamil nadu, 620015, India
Email: prof.m.anitha@gmail.com

**Funding information**
Department of Science and Technology, Government of India

In the present era, healthcare systems grapple with substantial volumes of medical data. However, a significant portion of this data is marked by incompleteness, inconsistency, errors, and unsuitability for training Machine Learning (ML) or Deep Learning (DL) algorithms. This necessitates preprocessing the data to render it amenable to utilization by ML/DL algorithms. Medical datasets predominantly feature two types of attributes: numerical and categorical values. The conversion of categorical features into numerical vectors is a crucial step in preparing the data for ML/DL algorithms, known as Feature Engineering (FE) based categorical encoding. Conventional and straightforward encoding of categorical features, termed one-hot encoding, generates multiple columns, thereby transforming data from a lower-dimensional to a higher-dimensional space. This approach poses challenges, including increased memory requirements due to the proliferation of columns. Considering these issues, this research proposes an encoding

**Abbreviations:** ABC, a black cat; DEF, doesn't ever fret; GHI, goes home immediately.

[*]Equally contributing authors.

technique named "Weight of Evidence with Entity Embedding" (WoEEE). The WoEEE approach bolsters the predictive capabilities of ML/DL algorithms by calculating the weight of evidence and concurrently mitigates dimensionality issues. To empirically validate the proposed method, it is tested on six diverse datasets: Breast Cancer, Hospital Readmission, Vadu, Covid-19, Stroke, and Heartstatlog. Four distinct ML/DL algorithms—Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), and a simple Feed-forward Neural Network (NN)— are employed for testing. The results obtained demonstrate that the WoEEE approach yields an average improvement of 11.18%, 10.37%, 5.83%, 7.58%, 7.83%, and 6.83% across all combinations of datasets, classifiers, and encoding methods. Furthermore, an Anova test is performed to confirm the effectiveness of WoEEE in encoding categorical data, especially for tasks involving binary classification. This enhances the treatment of categorical data in ML and data analytics scenarios. Overall, WoEEE shows potential as a valuable approach for categorical data encoding, making a positive contribution to the creation of effective techniques for handling this type of data in real-world applications.

**KEYWORDS**
Feature Engineering, Encoding, Entity Embedding, Categorical Data Encoding, Machine Learning

## 1 | INTRODUCTION

In the realm of Medical Data Science, each nation should harness its populace's medical data to glean insights for enhancing healthcare services. Medical data encompasses an individual's health status and medical interventions. Equipped with current medical data of patients, healthcare providers are poised to offer optimized, safer, and tailored care, leading to heightened efficiency and superior quality of service provision. Moreover, medical data serves as a valuable resource for researchers and research institutions in their pursuit of innovating novel treatments and medical devices. In light of this, the medical data essential for healthcare providers and researchers necessitates preprocessing to ensure its utility.

A considerable portion of medical data exhibits incompleteness, inconsistency, and inaccuracies. Consequently, subjecting such unprocessed data to statistical analysis poses risks and mandates appropriate preprocessing measures. The unprocessed medical data encompasses both numerical and categorical attributes, each of which can be

subjected to diverse transformation or encoding techniques. Within this context, numerous existing ML/DL models operate under the assumption that data and features inherently reside within a numerical domain, characterized by order and meaningful distances. This allows ML/DL models to consistently execute arithmetic operations, calculate central tendencies, dispersions, and measures of distance. However, categorical attributes do not universally conform to the aforementioned assumption, thereby rendering them incompatible with several mathematical operations and computations. Consequently, the conversion of categorical attributes into numerical equivalents becomes imperative. This process, involving the translation of categorical data/features into numerical representations, is commonly referred to as encoding [2].

Conventional techniques for encoding categorical features encompass a range of approaches such as one-hot, label, ordinal, binary, frequency, target, and mean encoding. Among these, one-hot encoding stands as the most recognized and widely used method by data analysts and scientists. By employing one-hot encoding to convert categorical attributes into numeric formats, the feature count expands correspondingly, contingent on the cardinality of the specific categorical dataset [8]. Consequently, this process gives rise to a substantial volume of sparse data for each categorical attribute. Additionally, due to the equidistant nature of feature vectors from one another, the inter-variable correlation would be forfeited [5]. Consequently, the decision has been made to introduce an encoding approach tailored to effectively manage categorical features. The aim of this proposed encoding technique is to retain the fundamental characteristics of categorical attributes while concurrently addressing memory utilization concerns. This transformative process is denoted as "Feature Engineering (FE) based categorical encoding." The FE-based categorical encoding method serves the purpose of assimilating data for learning purposes and extracting morphological insights during the training of ML algorithms.

The proposed approach amalgamates the encoding methodologies of Weight of Evidence and Entity Embedding. Although the devised Weight of Evidence (WoE) transformation is underpinned by a robust mathematical framework [26], its conceptual genesis is rooted in real-world scenarios. In our daily lives, we routinely make judgments based on the probability of certain events unfolding. While some situations and their corresponding decisions hold lesser significance, others of greater intricacy demand inputs from multiple sources. Irrespective of the complexity of a decision, its outcome's likelihood is often far from being a concrete reality, as it hinges on extensive data. Many of these data points might exhibit intricate interdependencies as elaborated in [6]. In this context, it becomes imperative to uncover the rationales behind each decision and discern the weight of the evidence. Essentially, this constructs a visualization of the risk associated with a specific choice or factual assertion along a linear continuum, facilitating data analysts and scientists in their risk assessment endeavors.

Similarly, the incorporation of the Entity Embedding (EE) technique [1] provides an additional advantage. This technique enhances the predictive prowess of classifiers through an exploration of the fundamental attributes of each independent variable within its embedding space. By opting for a predefined embedding size, the method reduces the multitude of distinct elements within categorical features into a compact vector representation of fixed dimensions. In doing so, it aligns similar values within the embedding space, thereby illustrating the inherent coherence of the data. This approach proves particularly adept at leveraging DL for the analysis of tabular data. Given that categorical data values inherently exhibit interdependencies, EE's capacity to align them in the embedding space holds notable significance.

As an illustration, let's consider the application of entity embedding to represent two distinct colors, namely "brown" and "black," within a given feature. Through this approach, two commensurate values emerge, with one value signifying the "shade" attribute and the other embodying the "primary color composition." This dual-value representation equips the model with an understanding of how each variable interacts, thereby facilitating more accessible learning and simultaneous performance enhancement. Consequently, the proposed model integrates the Weight of

Evidence encoding method with Entity Embedding and subsequently contrasts it with established benchmark encoding methods including label, one-hot, and binary encoding. To evaluate the effectiveness of the model, a selection of ML classifiers, such as Decision Tree, Random Forest, Logistic Regression, and Multi-layer Perceptron, are employed in the experimentation. The performance of each classifier is quantified in terms of accuracy. This paper underscores the significance of employing feature engineering-driven categorical encoding to accurately classify unprocessed medical data into appropriate class labels, thereby elevating classification accuracy.

The remainder of the paper is structured as follows: In Section 2, an extensive examination of existing literature and research pertaining to diverse categorical encoding methodologies for ML and DL is presented. Section 3 provides a comprehensive elucidation of the operational mechanics underlying the proposed Weight of Evidence with Entity Embedding (WoEEE) encoding scheme. Subsequently, Section 4 delves into the intricacies of the conducted experiments and offers an ANOVA analysis of the obtained results. The concluding remarks and potential directions for future research are outlined in Section 5.

## 2 | RELATED WORKS ON CATEGORICAL ENCODING IN ML/DL

Feature Engineering (FE) encompasses activities like feature selection, creation, generation, construction, and extraction. These techniques are commonly employed to enhance the concept of Feature Engineering. Besides enhancing classification accuracy through feature production, it's also essential to transform categorical features into numerical ones to make them comprehensible for ML/DL algorithms. The process of entity embedding for categorical variables, as introduced by Cheng Guo et al. in their work [1], involves mapping categorical variables to a function approximation. This approach is particularly prevalent in supervised learning neural networks. Unlike one-hot encoding, entity embedding is memory-efficient. The authors conducted a comparison between entity embedding and other ML methods like k-nearest neighbors, random forest, gradient boosting trees, and neural networks. Notably, neural networks exhibited exceptional performance in the context of entity embedding. To validate their approach, the authors conducted experiments using the Rossman Sale dataset from the Kaggle repository to predict daily sales. The study's ultimate inference was that adopting the entity embedding technique led to a reduction in the mean absolute percentage error, thereby affirming its efficacy.

Zhang et al. [2] proposed a novel method for transforming categorical features into numerical representation in which the numerical learning algorithm understands the categorical data's core properties. The authors studied the pairwise dissimilarity between categorical data and continuous embedding that uses manifold learning. The results reveal that the proposed method outperforms existing proven methodologies such as dummy and density-based encoding or algorithmic strategies such as Decision tree-based classification for many benchmark datasets. John T. Hancock et al. [3] published a survey on categorical data for neural networks. The three primary methods for implementing categorical data encoding are determined, algorithmic, and automatic techniques. The authors claim that the determined approaches are preferable for large datasets than algorithmic or automatic strategies since they require less computation. More study is needed for embedding techniques like EDLT and GEL.

Cerda et al. [4] discuss categorical data similarity encoding as well as one hot encoding for uncurated data. The dirty data is an issue, and datasets with high cardinality repeat themselves. The authors demonstrate that the similarity encoding technique enhances prediction using seven genuine datasets. Finally, the authors give the prediction score results of similarity encoding with a 3-gram distance. Wang et al. [6] discuss, the data pre-processing methods used in data mining. The data transformation procedure is used to identify the strategy employed for categorical data encoding. The authors find that the information probability of category symbols significantly boosts binary classification

accuracy compared to a frequency-based encoding technique.

In [7], Zdravevski et al. use the weight of evidence as an encoding approach, to solve the challenge of translating nominal data to numerical features. The key contribution is applying the WoE technique for multiclass classification problems. To do so, the technique must overcome the requirements of the basic description of the WoE parameters, such as positive and negative classes. The dataset must have just two classifications according to the weight of the evidence parameter. This is possible if the multiclass classification problem can be represented as a set of binary classification problems. A similar strategy, known as one-vs-all or one-vs-the-rest, has been used to generalize several ML algorithms that only accept two classes natively (e.g., support vector machines). The WoE transformation can be generalized for multiclass situations using the one-vs-all technique. The performance of the proposed model was compared between original data and transformed, proving better results with a neural network algorithm.

Cerda et al. aim to find a method for representing high-cardinality string categorical variables using a low-dimensional approach [8]. The objective is to make this approach scalable for a large number of categories, understandable for end users, and suitable for statistical analysis. The researchers present two techniques for encoding strings: one involves Gamma-Poisson matrix factorization based on substring counts, while the other employs a min-hash encoder to quickly estimate string similarity. The min-hash technique transforms set relationships into simpler inequality relationships. Both of these methods are designed to be scalable and adaptable to streaming data. The application of these strategies enhances supervised learning involving high-cardinality categorical variables, as indicated by assessments on both actual and simulated datasets. In conclusion, the study suggests that if scalability is a crucial factor, the min-hash encoder is preferable since it doesn't necessitate any data fitting. On the other hand, if interpretability is a primary concern, the Gamma-Poisson factorization is recommended, as it can be interpreted as a form of one-hot encoding applied to inferred categories, with meaningful feature names. Notably, neither of these models requires elaborate feature engineering or extensive data cleaning, and automated machine learning techniques can be employed for handling string inputs.

According to Lopez-Arevalo et al. [9] information preservation was validated when changing categorical variables to their numerical representation of vector space. The methods utilized were one-hot encoding and feature hashing, followed by data discretization for numerical values. The reduced overabundance of dummy features or spurious data, improved memory efficiency. The Land Change Modeler handbook by another author Eastman, J.R. [17] advises that categorical variables can be encoded using the idea of Population Evidence Likelihood (PEL). One of the K categories will be denoted by the letter k. The equation 1, explains the intersection of category k with the land change between two-time points, divided by the amount of the land change, which is the PEL for category k. The category with the most significant change magnitude is given the highest value by population evidence likelihood.

$$\text{Likelihood for category } k \text{ in the population} = \frac{\text{Size of change on category } k}{\text{Size of change}} \tag{1}$$

On the other hand, the Geomod land-change simulation model uses population empirical probability (PEP), also known as change intensity, to represent categorical variables. The equation 2, explains the intersection of category k with the land change between two time periods, divided by the size of category k yields the PEP for category k. The category with the most change intensity receives the highest PEP encoding value.

$$\text{Population Empirical Probability for category } k = \frac{\text{Size of change on category } k}{\text{Size of category } k} \tag{2}$$

A study by Swati Sachan et al. [22] takes a detailed look at the problem of categorical attributes confusing decision-making. In comparison to numerical or continuous attributes, categorical attributes are essentially non-numerical. The

inclusion of partial and ambiguous values in categorical qualities, as well as their fundamental non-numerical nature, adds to the ambiguity in decision-making. Three causes of uncertainty in categorical attributes have been identified in this work. The suggested technique in this study addresses informational uncertainty, unforeseen uncertainty in the decision task environment, and uncertainty in categorical attributes due to a lack of pre-modelling explainability. Several strategies for mapping from categorical to numerical space such as arithmetic operations and relevant distance measures have been discussed. However, these strategies are ineffective when it comes to studying the predictive value of categorical variables. Another issue to consider is that if the categorical variables have a more significant number of categories and the feature space expands, resulting in a high dimensional feature space and a substantial loss in classification accuracy. As a result of these issues, computing costs have grown, and memory efficiency has decreased.

While doing the transformation from categorical to numerical data which helps ML/DL algorithms understand the given data well, most of the existing works end up with the following drawbacks: (i) The predictive nature of categorical features is not learned, (ii) Monotonic relationship is lagging between the independent variables and dependent variables during the encoding process, and (iii) The increase in the number of columns, increases the need for memory to store the resulting columns. The issues mentioned above will be addressed appropriately in the proposed work. To accomplish the same, the proposed work has the following two-fold contributions:

- The proposed WoEEE method ensures optimal extraction of the predictive essence inherent in categorical features. Likewise, it aptly grasps this essence to mitigate the risk of subpar performance by ML/DL models, particularly in relation to the computational expenses associated with time and space.
- The introduced WoEEE approach generates a fixed-length vector representation for every categorical data/feature. This imposition curbs the proliferation of feature columns throughout the encoding procedure and concurrently establishes a monotonic connection between the encoded categorical variables and the response variable. This alignment consistently contributes to the enhancement of classification accuracy.

## 3 | THE PROPOSED METHODOLOGY

To gain a precise understanding of the importance of encoding categorical data, this section discusses the various aspects of the proposed method.

### 3.1 | Types of Categorical features

It is recognized that for each machine learning algorithm, the training features need to be represented in a numerical vector space. The inherent significance of the data becomes discernible only through numerical features. In this context, various types of features are categorized as nominal and ordinal. Nominal features are characterized by being grouped into distinct categories without any specific order among them. Nominal features lack numerical values and are sometimes referred to as "labeled" or "named" data. Examples of nominal data include personal names, hair colors, and genders. Moving on, ordinal features are data types that exhibit a predefined order or scale, but there isn't a universally standardized scale to quantify the variations between variables at each stage of the sequence. While often classified as categorical data, ordinal data demonstrates attributes of both categorical and numerical data, placing it in an intermediate realm. Its classification as categorical data primarily stems from its possession of more categorical characteristics. Instances of ordinal data encompass Likert scales, interval scales, bug severity ratings, and

customer satisfaction survey responses. While the techniques for collecting and analyzing data may differ across cases, these examples consistently fall under the category of ordinal data. The key attributes of categorical data encompass Categories, Qualitativeness, Analytical Properties, Graphical Analysis, Interval Scaling, Numeric Values, and Inherent Nature. These attributes are outlined in Table 1 to describe the characteristics of categorical data.

**TABLE 1** Characteristics of categorical data

| Characteristics | Description |
|---|---|
| Categorical | Categorical data is classified into two main types: nominal data and ordinal data. Nominal data, often recognized as named data, is a type of data employed for labeling items with distinct names. On the other hand, ordinal data exhibits a specific scale or order. |
| Qualitativeness | Categorical data falls under the category of qualitative data. In contrast to using numerical values, it employs a series of words or strings to depict an event or attribute. |
| Analysis | In the case of ordinal categorical data, both the median and mode can serve as measures of central tendency. While the mode is computable and interpretable for nominal categorical data, the mean and median might not always be calculable. Ordinal data is often subject to evaluation using univariate and bivariate statistics, regression analysis, assessments of linear trends, and the application of classification algorithms. |
| Graphical Analysis | Data visualization can be achieved through the utilization of either a bar chart or a pie chart. A bar chart is frequently employed to analyze frequencies, while a pie chart is commonly utilized to illustrate proportions or percentages. |
| Interval Scale | When dealing with ordinal data, which possesses a predetermined order, the scale may lack a clearly defined interval. This distinction doesn't apply to nominal data. |
| Numeric values | Despite the qualitative nature of categorical data, there are instances where numerical values can be present. However, these values lack quantitative attributes, and it's not possible to perform arithmetic operations on them. |
| Nature | Categorical data can be categorized as either binary or non-binary, depending on its inherent characteristics. A binary question entails two possible responses: "Yes" or "No." However, introducing an additional option like "Maybe" transforms it into a non-binary scenario. |

## 3.2 | Weight of Evidence with Entity Embedding

The Weight of Evidence (WoE) is used to assess a grouping strategy's "strength" in identifying the good from the bad [26]. The primary goal of this strategy is to establish a predictive model for estimating the target variable to know true and false class values, which provides predictive power. The WoE is a metric for determining the amount of evidence that supports or disproves a concept. WoE is calculated using the equation 3 given below:

$$WoE = \left[ ln\left( \frac{Distributionof PositiveClass}{Distributionof NegativeClass} \right) \right] * 100 \tag{3}$$

When the ratio of P(Positive Class) to P(Negative Class) equals 1, the Weight of Evidence (WoE) becomes 0,

suggesting a situation where the group's outcome is random. If P(Negative Class) is greater than P(Positive Class), the odds ratio is one, resulting in a negative WoE. Conversely, if P(Positive Class) outweighs P(Negative Class) within a group, the odds ratio remains one, and the WoE becomes positive. In the Probabilistic approach, the WoE method is a data-driven strategy that draws upon the Bayesian probability model, offering distinct advantages over alternative statistical techniques. Essential parameters for implementing WoE include positive weight (W+) and negative weight (W-). This method assesses the weight associated with each conditioning factor (B) based on the presence or absence of a class (A) within the group. This is outlined in equations 4 and 5, where P and ln represent probability and natural logarithm, respectively. B and ¬B signify the presence and absence of the conditioning factor, respectively.

$$W_i^+ = ln\frac{P(B|A)}{P(B|A)} \tag{4}$$

$$W_i^- = ln\frac{P(\neg B|A)}{P(\neg B|A)} \tag{5}$$

Also, A and ¬A represent the absence and presence of a group. A positive weight (W+) indicates the presence of the conditioning factor in the target class, and its positive class distribution indicates a positive link between the conditioning factor's presence and the occurrence of the class variable. A negative weight (W-) reflects the level of negative association and implies the lack of the conditioning factor. The WoE value only tells how confident the feature will help to predict an event's probability correctly. Many studies have used the WoE method for credit card fraud detection and also in the field of flood susceptibility mapping [26]. However, this method is relatively new in categorical data encoding in the healthcare domain.

WoE is better than one-hot encoding as one-hot encoding will have to create h-1 new features to accommodate one categorical feature with h values. This implies that the underlying model has no need to approximate h-1 coefficients rather than approximating one single coefficient in the case of the Logistic Regression algorithms, where $b_i$ stands for coefficients of the features to be determined. However, in WoE variable transformation, the weights were normalized to 0 and 1. Conditioning factors were reclassified based on these normalized values and consequently fed into the Entity Embedding (EE) encoding model. The normalization should be done because the weights of conditioning factors vary in dimensions and are not appropriate for direct input for the ML classifier model.

Embedding is the mapping of a categorical variable [1] to an n-dimensional vector in formal terms. These benefits in two ways: to begin, it limits the number of columns required for each category and second, by its very nature, embedding naturally groups comparable variables together. Assume to use the weekday as a feature in our neural network. Start the tensor by creating a 7x4 matrix that maps a day of the week to each row. After that, we replace a specific weekday with its associated vector. Now, this matrix can be used to find non-linear correlations between variables. Embedding turns a day of the week into a four-dimensional numerical space instead of one-hot encoding, which can only be a single value and it has semantic value after training the model. Saturday and Sunday, for example, maybe more closely related than Saturday and Wednesday. It is essential to put corresponding values of a categorical variable closer together in the embedding space with entity embedding [1, 2, 6]. Entity embedding is closely connected to the embedding of a finite metric space problem in topology and uses a real number to define the similarity of values.

$$d(x_i^p, x_i^q) = \langle |f(x_i{}^p, \bar{x}_i) - f(x_i^q, \bar{x}_i)| \rangle_{x_i} \tag{6}$$

$$d(x_i^p, x_i^q) = 0 \leftrightarrow x_i^p = x_i^q \tag{7}$$

$$d(x_i^p, x_i^q) = d_i(x_i^q, x_i^p) \tag{8}$$

$$d(x_i^p, x_i^r) \leq d(x_i^p, x_i^q) + d(x_i^q, x_i^r) \tag{9}$$

Moving forward, in the equation for function approximation (equation 6), a finite metric space $(C_i, d_i)$ is established for each categorical variable $x_i$, wherein $C_i$ signifies the set encompassing all possible values of $x_i$. The distance function denoted as $d_i$ quantifies the dissimilarity between any two value sets $(x_{pi}, x_{qi})$ of $x_i$, with $d_i$ representing the metric applied within $C_i$. The parameter $d_i$ essentially measures the similarity between the two value sets $(x_{pi}, x_{qi})$. There are various approaches to define this parameter, with one particularly straightforward and intuitive method [1]. Equations 7 and 8 may not be inherently valid in real-world scenarios if distinct values consistently yield the same output. However, this implies redundancy in one of the values, and a resolution can be reached by consolidating these values into one, effectively redefining the categorical variable to formulate equation 9. The objective of entity embedding involves the transformation of discrete values into a multi-dimensional space, where values yielding similar function outputs are positioned in proximity. Achieving an optimal solution for representing categorical variables would entail utilizing fewer dimensions than the count of distinct categories, while also ensuring that akin categories are situated closer to each other.

$$Similarity(A, B) = cos(\theta) = \frac{A \cdot B}{\|A\| \, \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{10}$$

In equation 10, the cosine of two non-zero vectors can be derived by using the Euclidean dot product formula: Given two vectors of attributes, $A$ and $B$, the cosine similarity, $cos(\theta)$, is represented using a dot product and magnitude as where $A_i$ and $B_i$ are components of vector $A$ and $B$, respectively. The resulting similarity ranges from 0 to 1, and if the cosine value of two vectors is close to 1, then it indicates that they are almost similar. A zero value indicates that they are dissimilar or not correlated.

## 3.3 | Preliminary processing

In this section, we will be discussing the proposed WoEEE encoding approach for categorical data which incorporates both weight of evidence encoding and entity embedding methods. The encoding is done in three stages that involve cleaning and pre-processing the data, handling class imbalance, and applying the WoEEE encoding scheme. To better understand the proposed approach, an architecture diagram is presented in Figure 1 and a step-by-step explanation of the encoding approach is shown. The benchmark datasets are taken from the UCI Machine Learning Repository and INDEPTH Data Repository for experimentation is summarized in Table 2:

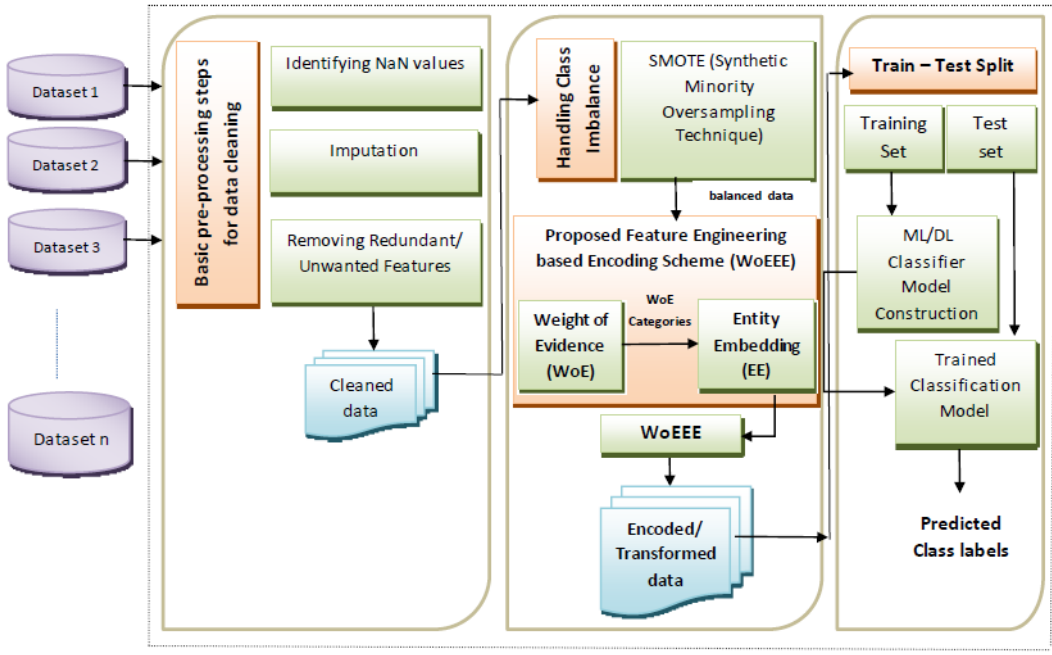


**FIGURE 1** The architecture diagram of the proposed WoEEE encoding scheme

### 3.3.1 | Dataset introduction

The proposed architecture initiates with accepting the raw medical data as input. The input is provided with necessary notation. Consider D is a numerical and categorical type dataset containing N number of d-tuples of the form ($f_i1$, $f_i2$, $f_i3$....., $f_id$) representing feature vector that contains numerical and categorical values. These vectors can be arranged in a N x d matrix of the form:

$$D = \begin{bmatrix} f_{11} & f_{12} & \dots & f_{1d} \\ f_{21} & f_{22} & \dots & f_{2d} \\ . & . & \dots & . \\ . & . & \dots & . \\ . & . & \dots & . \\ f_{N1} & f_{N2} & \dots & f_{Nd} \end{bmatrix}$$

Let D [,j] be the jth column of D containing the values of the jth feature(f) (with j=1,2,3,..d). Now, we can assume numerical features in X are valued in R, and categorical features are valued as non-empty strings that satisfy the regular expressions A-Z, a-z, 0-9 +. With the above mentioned assumption, for each D [,j] containing categorical values, map each unique instance or observation to an integer value. At this instant, D will contain numerical values representing both categorical and numerical features.

**TABLE 2** Dataset description

| Dataset | Dataset Sources | # Features | # Numerical | # Categorical | # Instances | # Classes |
|---|---|---|---|---|---|---|
| Breast Cancer | UCI machine Learning Repository | 10 | 1 | 9 | 201 | 2 |
| Hospital Readmission | UCI machine Learning Repository | 50 | 15 | 35 | 101766 | 2 |
| Vadu | INDEPTH Data Repository | 47 | 20 | 27 | 18426 | 2 |
| Covid-19 | Central Disease Control(CDC) | 10 | 0 | 9 | 20000 | 2 |
| Stroke | UCI machine Learning Repository | 12 | 7 | 5 | 5110 | 2 |
| HeartStatlog | UCI machine Learning Repository | 13 | 7 | 6 | 270 | 2 |

### 3.3.2 | Pre-processing for data cleaning

In the context of our experimentation, we are working with six distinct datasets, all of which contain instances of NaN (Not a Number) values. These NaN instances are denoted by the presence of '0' or '999' entries within the dataset. These missing values can potentially hinder the seamless operation of ML/DL algorithms, which require complete and coherent data inputs. To address this issue, we've outlined two primary strategies: the first involves eliminating NaN values through either dropping them from the dataset, while the second entails imputing these NaN values with appropriate substitutes, ensuring that the subsequent computations remain meaningful and accurate for ML/DL algorithms.

The decision of whether to drop or impute NaN values is contingent on the extent of their prevalence within each dataset. Specifically, we calculate the percentage of missing values in each dataset. If this percentage falls below the 15% threshold, we opt for the removal of NaN values from the dataset. This is because a relatively low percentage of missing data is unlikely to significantly skew the overall dataset's integrity.Conversely, if the proportion of missing values exceeds the 15% threshold, a more nuanced approach is required. In such cases, discarding the NaN values outright could lead to a loss of valuable information. Instead, we turn to traditional imputation techniques to intelligently fill in these gaps. These techniques include:

- Forward Fill (or Next Value Imputation): This method involves substituting the NaN value with the next available non-NaN value in the sequence. It is particularly suitable when the data follows a certain chronological or sequential order.
- Backward Fill (or Last Value Imputation): Similar to forward fill, backward fill replaces the NaN value with the last non-NaN value in the sequence. It's useful for time-series or ordered data.
- Mean Imputation: NaN values are replaced with the mean of the non-NaN values within the same feature. This method can help maintain the overall statistical properties of the dataset.
- Mode Imputation: In this technique, the NaN values are substituted with the mode, i.e., the most frequently occurring value, within the same feature. Mode imputation is suitable for categorical data.

These imputation techniques ensure that the NaN values are substituted with relevant and contextually appropriate values, maintaining the integrity of the dataset. By adopting a thoughtful approach based on the missing values' extent and considering established imputation methods, we aim to prepare the datasets for meaningful analysis and effective employment of ML/DL algorithms.

### 3.3.3 | Handling Class imbalance (SMOTE)

Once data cleaning has been successfully carried out through appropriate pre-processing, the subsequent task involves addressing class imbalance within each dataset. When dealing with imbalanced datasets, a common challenge arises where the majority of machine learning techniques tend to overlook the minority class(es), resulting in subpar performance, even though the performance of the minority class often holds significant importance. To combat this issue, oversampling the minority class presents itself as a potential solution. The Synthetic Minority Oversampling Technique (SMOTE) is a method that can be employed to effectively tackle class imbalance. While duplicating instances within the minority class is a straightforward approach, these duplicated examples might not provide novel insights to the model. Instead, the creation of new instances by combining existing ones proves more beneficial.

The functioning of SMOTE involves the selection of instances within the feature space that are in close proximity. A line is drawn in the feature space between these instances, and a new sample is generated along that line. More precisely, the process begins with the random selection of an instance from the minority class. Subsequently, the k closest neighbors are identified (typically with k=5). Among these neighbors, one is randomly chosen, and a synthetic example is crafted at a randomly determined position in the feature space between the two selected samples. This methodology can generate numerous synthetic instances as required for the minority class. As mentioned in [25], an additional technique known as random under sampling can be employed to decrease the number of examples within the majority class.

### 3.3.4 | Proposed WoEEE Encoding Scheme

After finishing the preliminary processing, in the next phase, Weight of Evidence (WoE) for every categorical data of all six datasets are calculated. WoE is a technique used to encode categorical variables for classification. The WoE measures the predictive power of an independent variable in relation to the dependent variable. It provides understanding relationships between important independent variables and the dependent variable. When using WoE encoding method for categorical and numerical data, there are some rules to be followed. For categorical data, generally, each category/value is a bin. Smaller categories are grouped together. Numerical data are split into categories. Binning or grouping values of both categorical and numerical data is done with subsequent rules: To begin, each bin should contain at least 5% of the observations. Second, for both positive/true class (1) and negative/false class (0), each bin must be non-zero. Third, WoE should be monotonic, that is, it should either increase or decrease with the groupings.

WoE is calculated by taking the natural logarithm (log to base e) of division of percentage of false class events and percentage of true class events. The depicted mathematical equation of WoE is already shown in equation 3. During the calculation of WoE, to ensure each bin has non-zero values, the values of each categorical data (x) is exposed along with their counts C(x), to know if any zero values are present. If zero values exist, those values must be cleared by removing that value before proceeding with the calculation. Because it never gives any information about the categorical data, when it is related with target or class variable. If the WoE value of a category is positive, then it means that the distributions of true class events are greater than the distribution of false class events. If it is a negative WoE value, then the distribution of true class events are lesser than the distribution of false class events. Ultimately, the derived WoE values describe, how certain independent variables influence the dependent variables.

Then, the obtained positive and negative WoE values of each categorical data calculated earlier is fed as input to Entity Embedding encoding method. It is a vector representation of an entity or categorical data. The Entity Embedding is used in the place of one-hot encoding, since it can map related values closer together in embedding

space, revealing the inherent continuity of the data. On the other hand, one-hot encoding ignores informative relations between feature values. To start with the process of embedding categorical data, the output of weight of evidence is considered as object type and the value counts of each categorical variable are listed. Now the training data consists of both categorical and numerical data.

Then the choice of embedding dimension is optional, essentially, it is a hyperparameter that one needs to choose beforehand and investigate. One rule of thumb is to choose half of the cardinality (n) of the categorical feature if n<= 50 in length. That is to find the embedding size of any feature's category, the number of unique categories should be divided by two and it is taken as the embedding size of every category. The training data should be given to different ML algorithms such as DT, RF, LR, and NN. Every dataset has gone through a comparison with various benchmark encoding methods such as Label encoding, One-hot encoding, Binary encoding, and the proposed encoding approach (WoEEE).

**Algorithm 1**: Feature Engineering based Categorical encoding – Weight of Evidence with Entity Embedding Encoding scheme

1. FOR every categorical feature (CF) and numerical feature (NF)present in DS DO
   - Cleaned_data = PRE-PROCESSING(DS)

   END FOR
2. IF there exists class imbalance in the Cleaned_data THEN
   - balanced_data = SMOTE(random_state = 20)

   END IF
3. FOR every categorical independent feature i in balanced_data DO
   - Weight of Evidence balanced_data
   - A = Calculate ratio of proportion for dependent variable False class (Y=0) of independent variables(X)
   - B = Calculate ratio of proportion for dependent variable True class (Y=1) of independent variables (X)
   - $WoE_{x=i} = Ln(A/B)$

     $WoE_{categories}$=WoE$_{x=i}$

   END FOR
4. FOR i in $WoE_{categories}$ DO
   - $Entity\_Embeddingsize$ <—- $n\_cat + 1(totalnumberofcategories + 1)1ton\_cat$
   - M<—- number of instances or records of dataset
   - n_cat_lists M, Entity_Embedding_size
   - n_other_lists<—-other numeric columns
   - output_vector_representations_woe_categories n = i
   - encoded/transformed data <—— output_vector_representations_woe_categories

   END FOR
5. FOR i in encoded/transformed data DO
   - encoded_data_train, encoded_data_test = Split(encoded_data) // 80:20 split
   - Classifier model construction encoded_data_train
   - Trained constructed Modelencoded_data_test
   - Predicted class labels<—- Trained constructed Modelencoded_data_test

   END FOR

The complete processing of the proposed work is shown and described through the Algorithm 1. This algorithm takes training data (encoded data) and test data (encoded data) with two class labels as input. The output obtained

is predicted class labels (y1, y2, ..) for test data(encoded data). In step 1, data preprocessing for every dataset is performed and in step 2, imbalanced data is also handled using the method SMOTE. In step 3, the Weight of Evidence for every categorical data is calculated. In step 4, the output of the previous step is given as input to the Entity Embedding method and framing the embedding layer with the embedding size and unique values of categorical variables. Finally, in step 5, the predictions are made by all classifiers, and performance is evaluated using suitable performance evaluation metrics.

## 3.4 | Experimental Results and Discussions

A concise and comprehensive description of each ML model, including its purpose, key features along with model training, evaluation, interpretation, and reproducibility are discussed in this section. The experiment was designed to evaluate the performance of the proposed WoEEE encoding approach with ML models on publicly available six medical datasets. The datasets consist of minimum 286 samples and maximum 101763 samples, with minimum 9 features and 47 maximum features. Table 3 presents a comprehensive overview of the datasets analyzed in this research, including relevant information such as the number of observations, categorical and numerical features with cardinality of each feature. The datasets are pre-processed to remove missing values, unwanted features and split into training (80%) and testing (20%) sets. The categorical variables in the datasets are encoded using one-hot encoding, label encoding, binary encoding, and the proposed WoEEE encoding approach.

The encoded data was then used to train various predictive models, including DT, LR, RF and NN. The performance of each model was evaluated using accuracy. The models' performance is evaluated using anova analysis. The results are compared with several state-of-the-art models to demonstrate the effectiveness of the proposed WoEE. The model is also subjected to a Wilcoxin Test to study the impact of encoding approach on its performance.

**TABLE 3** Description of the datasets used

| Dataset | # Numerical features | # categorical features | #cardinality | #Classes |
|---|---|---|---|---|
| Breast Cancer | 1 | 9 | 38 | 2 |
| Hospital Readmission | 15 | 35 | 117 | 2 |
| Vadu | 20 | 27 | 88 | 2 |
| Covid-19 | 0 | 9 | 23 | 2 |
| Stroke | 7 | 7 | 10 | 2 |
| Heartstatlog | 1 | 9 | 19 | 2 |

### 3.4.1 | Building ML models

**Decision Trees** are popular algorithms used in machine learning for both classification and regression tasks. When working with categorical data, it is important to encode the variables into numerical values before building the model. Hyper-parameters play a crucial role in determining the performance of a Decision Tree model, especially when working with categorical data. Some of the hyperparameters for categorical data encoding in Decision Trees include cri-

terion, splitter, max_depth, min_samples_split, and min_samples_leaf. The criterion hyperparameter determines the function used to measure the quality of a split, such as "Gini" or "Information Gain". The splitter hyper-parameter controls the strategy used to choose the split at each node, such as "best" or "random." The max_depth hyperparameter sets the maximum allowed depth of the tree, while the min_samples_split and min_samples_leaf hyper-parameters set the minimum number of samples required to split a node and the minimum number of samples required to be at a leaf node, respectively. By tuning these hyperparameters, we can control the complexity and size of the Decision Tree model, avoiding overfitting and ensuring a good fit for the data. The code to implement Decision Tree model in this experiments are shown here.

```
model = DecisionTreeClassifier()
```

**Random Forest** is a popular machine learning algorithm that can be used for both regression and classification problems. It uses multiple decision trees to make predictions and combines the them to produce a final prediction. The hyper-parameters to consider when using a Random Forest Classifier for categorical data encoding are total number of estimators 20, with maximum depth of 28 and the criterion used is Gini index and minimum sample split of 15. The code to implement Random Forest classifier model in the experiments are shown here.

```
model = RandomForestClassifier(n_estimators, random_state)
```

**Logistic Regression** Logistic Regression is a popular machine learning algorithm used for binary classification (when the target variable has only two possible outcomes). When dealing with categorical data encoding, it is common to encode the categorical variables into numerical values using techniques such as one-hot encoding, ordinal encoding, or target encoding. The logistic regression model uses a linear equation to model the relationship between the independent variables (including the encoded categorical variables) and the dependent binary variable. The model outputs a probability of belonging to each class, and a threshold is applied to make the final binary classification. The coefficients of the independent variables are learned during the training process using maximum likelihood estimation. Logistic regression is a useful tool for encoding categorical data in classification tasks, especially when the relationship between the variables is relatively simple and linear. The code to implement Logistic regression classifier is shown here.

```
model = LogisticRegression (penalty='l2', solver='lbfgs', max_iter=1000,
class_weight='balanced')
```

**Neural Network** to model the relationship between the one-hot encoded categorical data, Label encoded data, binary encoded data, proposed WoEEE encoded data and the target variable, and we used a fully-connected neural network with three hidden layers. Each hidden layer consisted of 128 neurons and used the ReLU activation function. The output layer used a sigmoid activation function, as the target variable is binary in nature. To improve the results, dropout is used in the layers (0.3 and 0.2). The model is trained using binary cross-entropy as the loss function and the Adam optimization algorithm. The training process is performed using mini-batch gradient descent, with a batch size of 128 and 50 epochs.

```
Define the neural network model
model = Sequential()
model.add(Dense(128, activation='relu', input_shape=(x_train.shape[1],)))
model.add(Dense(128, activation='relu'))
model.add(Dense(128, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
Compile the model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

The experiments were carried out to prove that the encoded data using our proposed encoding approach which

guarantees the predictive nature of categorical features and to reduce the increase in the number of features during encoding process with improvement in classification accuracy. For experimentation, the Algorithm 1 was implemented in Python version 3.8 and was executed on a computer with Intel R Core TM i7 processor, and 16 GB of RAM.

## 3.5 | Evaluation Metrics

Binary classification is a common task in ML and involves predicting one of two possible outcomes. In order to evaluate the performance of our binary classification model, we used several commonly used evaluation metrics. To begin with, we calculated the accuracy, which measures the proportion of correct predictions made by the model. However, accuracy alone can be misleading, especially when the classes are imbalanced. Therefore, we also computed precision and recall, which respectively measure the proportion of correct positive predictions and the proportion of actual positive instances that were correctly identified by the model. Precision is important when the cost of false positives is high, while recall is important when the cost of false negatives is high. We also calculated the F1-score, which is the harmonic mean of precision and recall and is useful when both measures are important for the problem at hand.

In addition, we computed the area under the receiver operating characteristic (ROC) curve, which is a popular metric that considers the tradeoff between true positive rate and false positive rate for different classification thresholds. The ROC curve plots the true positive rate (sensitivity) against the false positive rate (1 - specificity) for different threshold values, and the area under the curve (AUC) measures the overall performance of the model regardless of the threshold value. AUC values range from 0.5 (random guessing) to 1.0 (perfect classification). Overall, we found that our model performed well, achieving high accuracy and F1-score, as well as a high AUC.

## 3.6 | Proposed WoEEE encoding method's Classification performance with the baseline models

This section presents the quantitative experimental results of the proposed work, wherein the role of the encoding procedure in classification tasks using various ML algorithms has been verified. The main objective of the WoEEE encoding approach is to improve the performance exhibited by the classification algorithms by extracting the predictive nature of categorical features from the mentioned datasets. This extraction of predictive nature is quantified in terms of the accuracy of classification, and it is argued that this accuracy is a consequence of the encoding's ability to retain the predictive nature of categorical data to discriminate the instances belonging to different classes. It is observed that proposed WoEEE encoding method is as good as one of the most popular encoding approaches in terms of predictive nature, but remarkably superior in terms of memory efficiency. This is achieved by providing a fixed vector representation for categorical data, which restricts the increase in feature columns. The significant observation to be noted is that, the proposed WoEEE encoding approach provides the best results in classification accuracy in all ML/DL algorithms, when compared with other benchmark encoding techniques.

Additionally, in connection with the above observation, the percentage improvement of the classification accuracy is a critical measure of the performance of a classifier. In this study, the effectiveness of a ML algorithm on six distinct datasets: Breast Cancer, Hospital readmission, Covid, Vadu, Stroke and HeartStatlog are evaluated. The (baseline) benchmark encoding method's accuracies of the classifiers (DT, RF, LR and NN) are determined by running it on each dataset before any modifications are made. After training and testing the classifiers on each dataset, the accuracy was recorded and compared to the (baseline) benchmark encoding method's accuracy to determine the percentage improvement. In Table 4, the classification accuracy results between label, one-hot, binary and WoEEE encoding method with the percentage improvement of classification accuracies are shown. The table also includes the average

accuracy of state-of-the-art models and the proposed WoEEE method's accuracy.

Overall, the proposed WoEEE encoding method outperformed the other benchmark encoding methods in all datasets, resulting in significant improvements in accuracy for all classifiers. For example, in the Breast Cancer dataset, the Decision Tree classifier achieved an accuracy of 73% with WoEEE encoding, which is an 11.33% improvement over the benchmark Label Encoding method. Similarly, in the Hospital Readmission dataset, the Neural Network classifier achieved an accuracy of 91% with WoEEE encoding, which is an 18.33% improvement over the benchmark Binary Encoding method.

The results also suggest that different encoding methods may perform differently for different datasets and classifiers. For example, in the Covid dataset, the One Hot Encoding method performed better than the other methods for the Logistic Regression classifier, but worse than the other methods for the Neural Network classifier. In summary, the proposed WoEEE encoding method shows promising results in improving the accuracy of classifiers for various datasets. The results also highlight the importance of carefully selecting the encoding method for a given dataset and classifier. The graphical representation of classification accuracy is shown in Figure 2.a, 2.b, 2.c, 2.d, 2.e and 2.f.

Subsequently, the prediction results of four classifiers namely Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), and Neural Network (NN), along with the performance of the classifiers are evaluated using several evaluation metrics, including Accuracy, Precision, Recall, F1-score, and ROC are presented in Tables 5, 6, 7, and 8 for six datasets (BC, HR, COV, VADU, STR, and HS). The datasets have been encoded using proposed WoEEE encoding method and three benchmark encoding methods. These tables provide a comprehensive comparison of the performance of the classifiers across different encoding methods and datasets.

Based on the comparison of the above Tables 5, 6, 7, and 8, it can be observed that the proposed WoEEE encoding method generally improves the performance of the classification models for all datasets compared to the benchmark encoding methods. Specifically, for the BC dataset, the proposed encoding method leads to improvements in all performance metrics for all classification models. The DT, RF, and NN models have the most significant improvement in accuracy, precision, recall, and F1-score, while LR has the most significant improvement in ROC-AUC. For the HR dataset, the proposed encoding method leads to improvements in most performance metrics for all classification models. The DT, RF, and LR models have the most significant improvement in accuracy, precision, recall, and F1-score, while NN has the most significant improvement in ROC-AUC.

For the COV dataset, the proposed encoding method leads to improvements in most performance metrics for all classification models. The DT and RF models have the most significant improvement in accuracy, precision, recall, and F1-score, while LR and NN have the most significant improvement in ROC-AUC. For the VADU dataset, the proposed encoding method leads to improvements in most performance metrics for all classification models. The DT, RF, and LR models have the most significant improvement in accuracy, precision, recall, and F1-score, while NN has the most significant improvement in ROC-AUC. For the STR dataset, the proposed encoding method leads to improvements in most performance metrics for all classification models. The DT, RF, and LR models have the most significant improvement in accuracy, precision, recall, and F1-score, while NN has the most significant improvement in ROC-AUC.

For the HS dataset, the proposed encoding method leads to improvements in most performance metrics for all classification models. The DT and RF models have the most significant improvement in accuracy, precision, recall, and F1-score, while LR and NN have the most significant improvement in ROC-AUC. Overall, the proposed WoEEE encoding method appears to be a promising approach to encoding categorical variables for classification problems, leading to improved performance for various classification models and datasets. Additionally, the classification algorithms of encoding methods for six datasets are explored in the Figure 2.a, 2.b, 2.c, 2.d, 2.e, and 2.f.

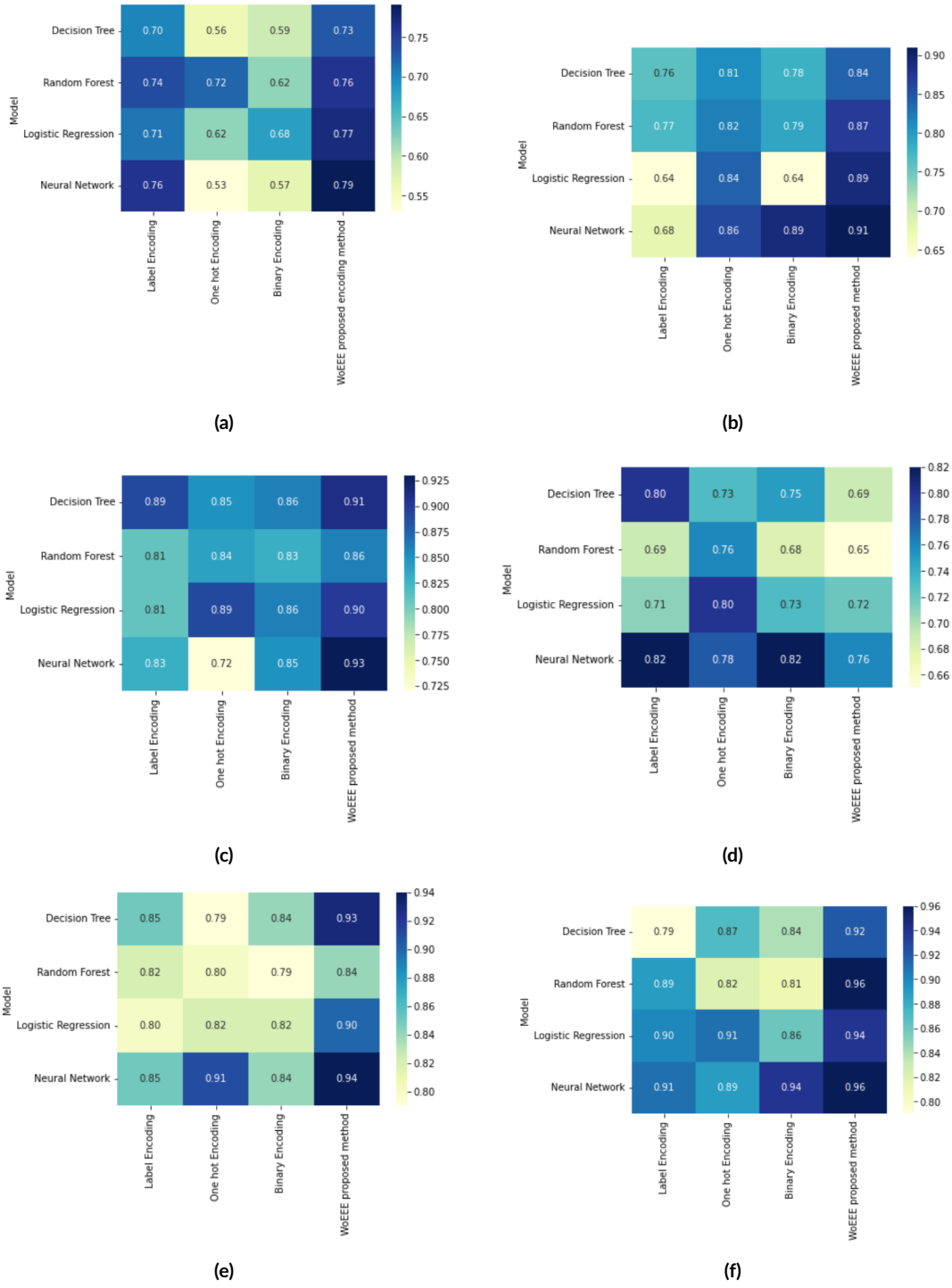


**FIGURE 2** **(a)** Exploring the performance of classification algorithms of Encoding Methods for Breast Cancer dataset. **(b)** Investigating the efficacy of classification algorithms across various encoding methodologies for a Hospital Readmission Dataset. **(c)** Investigating the classification algorithm performance across diverse encoding methodologies applied to a Covid Dataset. **(d)** Analyzing the efficacy of classification algorithms in the context of various encoding methods for the Vadu Dataset. **(e)** Investigating the classification algorithm performance with respect to different encoding techniques applied to the Stroke Dataset. **(f)** Undertaking an empirical analysis of classification algorithm effectiveness in relation to diverse encoding methodologies applied to the Stroke Dataset.

**TABLE 4** Evaluation of classification performance across multiple datasets, quantifying the percentage improvement achieved

| Dataset | Encoding methods | Decision Tree | Random Forest | Logistic Regression | Neural Network |
|---|---|---|---|---|---|
| Breast Cancer | Label Encoding | 0.7 | 0.74 | 0.71 | 0.76 |
| | One hot Encoding | 0.56 | 0.72 | 0.62 | 0.53 |
| | Binary Encoding | 0.59 | 0.62 | 0.68 | 0.57 |
| | Average of state-of-art models | 0.62 | 0.69 | 0.67 | 0.62 |
| | WoEEE proposed method | 0.73 | 0.76 | 0.77 | 0.79 |
| | % of improvement | 11.33 | 6.67 | 10.00 | 16.71 |
| Hospital Readmission | Label Encoding | 0.76 | 0.77 | 0.64 | 0.68 |
| | One hot Encoding | 0.81 | 0.82 | 0.84 | 0.86 |
| | Binary Encoding | 0.78 | 0.79 | 0.64 | 0.89 |
| | Average of state-of-art models | 0.78 | 0.79 | 0.71 | 0.81 |
| | WoEEE proposed method | 0.84 | 0.87 | 0.89 | 0.91 |
| | % of improvement | 5.67 | 7.67 | 18.33 | 9.80 |
| Covid | Label Encoding | 0.89 | 0.81 | 0.81 | 0.83 |
| | One hot Encoding | 0.85 | 0.84 | 0.89 | 0.72 |
| | Binary Encoding | 0.86 | 0.83 | 0.86 | 0.85 |
| | Average of state-of-art models | 0.87 | 0.83 | 0.85 | 0.80 |
| | WoEEE proposed method | 0.91 | 0.86 | 0.9 | 0.93 |
| | % of improvement | 4.33 | 3.33 | 4.67 | 13.00 |
| Vadu | Label Encoding | 0.79 | 0.89 | 0.9 | 0.91 |
| | One hot Encoding | 0.87 | 0.82 | 0.91 | 0.89 |
| | Binary Encoding | 0.84 | 0.81 | 0.86 | 0.94 |
| | Average of state-of-art models | 0.83 | 0.84 | 0.89 | 0.91 |
| | WoEEE proposed method | 0.92 | 0.96 | 0.94 | 0.96 |
| | % of improvement | 8.67 | 12.00 | 5.00 | 4.67 |
| Stroke | Label Encoding | 0.85 | 0.82 | 0.8 | 0.85 |
| | One hot Encoding | 0.79 | 0.8 | 0.82 | 0.91 |
| | Binary Encoding | 0.84 | 0.79 | 0.82 | 0.84 |
| | Average of state-of-art models | 0.83 | 0.80 | 0.8 | 0.87 |
| | WoEEE proposed method | 0.93 | 0.84 | 0.9 | 0.94 |
| | % of improvement | 10.33 | 3.67 | 10 | 7.33 |
| Heart Statlog | Label Encoding | 0.8 | 0.73 | 0.75 | 0.69 |
| | One hot Encoding | 0.69 | 0.76 | 0.68 | 0.65 |
| | Binary Encoding | 0.71 | 0.8 | 0.73 | 0.72 |
| | Average of state-of-art models | 0.73 | 0.76 | 0.72 | 0.69 |
| | WoEEE proposed method | 0.82 | 0.78 | 0.82 | 0.76 |
| | % of improvement | 8.7 | 1.67 | 10.00 | 6.99 |

**TABLE 5** Label Encoding classification performance results

| Dataset/ | DT | | | | | RF | | | | | LR | | | | | NN | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Label Encoding | Acc | Prec | Rec | F1-sc | Roc | Acc | Prec | Rec | F1-sc | Roc | Acc | Prec | Rec | F1-sc | Roc | Acc | Prec | Rec | F1-sc | Roc |
| BC | 70.01 | 46.15 | 42.86 | 44.44 | 65.15 | 74.36 | 54.55 | 42.86 | 47.97 | 70.12 | 71.18 | 60.04 | 42.85 | 50.43 | 71.42 | 76.16 | 63.64 | 50.57 | 56.23 | 73.34 |
| HR | 76.16 | 82.32 | 49.94 | 62.16 | 83.45 | 77.32 | 82.41 | 50.70 | 62.77 | 83.25 | 64.75 | 71.15 | 62.31 | 58.49 | 56.58 | 68.78 | 53.16 | 23.31 | 44.7 | 64.52 |
| COV | 89.13 | 75.01 | 82.90 | 57.14 | 78.70 | 81.31 | 80.11 | 43.90 | 47.30 | 78.86 | 87.26 | 79.91 | 57.52 | 54.12 | 70.39 | 83.57 | 49.1 | 52.68 | 56.34 | 80.76 |
| VADU | 79.91 | 86.64 | 88.32 | 87.47 | 72.13 | 89.64 | 89.15 | 84.38 | 80.15 | 86.77 | 90.12 | 83.49 | 86.26 | 80.12 | 83.43 | 91.42 | 88.54 | 74.31 | 80.8 | 84.57 |
| STR | 85.11 | 88.77 | 89.6 | 89.18 | 89.18 | 82.94 | 72.53 | 76.46 | 74.44 | 84.32 | 79.51 | 77.54 | 81.88 | 79.66 | 80.54 | 84.56 | 83.73 | 93.63 | 88.41 | 87.68 |
| HS | 80.25 | 73.10 | 84.38 | 88.52 | 88.74 | 73.25 | 84.85 | 87.50 | 86.15 | 85.13 | 75.23 | 87.13 | 84.38 | 85.71 | 85.29 | 69.16 | 77.51 | 79.14 | 78.31 | 80.85 |

**TABLE 6** One-hot Encoding classification performance results

| Dataset/ | DT | | | | | RF | | | | | LR | | | | | NN | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| One-hot Encoding | Acc | Prec | Rec | F1-sc | Roc | Acc | Prec | Rec | F1-sc | Roc | Acc | Prec | Rec | F1-sc | Roc | Acc | Prec | Rec | F1-sc | Roc |
| BC | 56.18 | 58.33 | 50.1 | 53.85 | 71.42 | 72.12 | 71.43 | 35.71 | 47.61 | 76.21 | 62.54 | 52.13 | 35.71 | 41.66 | 70.9 | 53.18 | 62.31 | 42.85 | 56.31 | 74.39 |
| HR | 81.32 | 81.49 | 48.15 | 60.53 | 82.87 | 82.07 | 79.98 | 49.37 | 61.05 | 82.96 | 84.37 | 71.86 | 69.01 | 59.64 | 57.75 | 86.18 | 53.51 | 49.95 | 91.38 | 69.55 |
| COV | 85.21 | 42.85 | 79.64 | 36.69 | 77.91 | 84.25 | 42.85 | 79.64 | 34.48 | 78.74 | 89.21 | 75.49 | 69.81 | 71.8 | 72.59 | 72.25 | 75.45 | 79.31 | 50.87 | 79.66 |
| VADU | 89.48 | 75.49 | 73.15 | 74.3 | 80.15 | 86.16 | 81.56 | 79.85 | 80.7 | 82.39 | 85.46 | 83.72 | 78.45 | 81.23 | 85.79 | 80.19 | 76.41 | 70.15 | 73.15 | 76.15 |
| STR | 79.29 | 77.69 | 82.31 | 79.93 | 79.28 | 96.42 | 96.26 | 96.6 | 96.43 | 96.42 | 79.29 | 77.69 | 82.31 | 79.93 | 79.28 | 94.41 | 92.14 | 97.1 | 94.55 | 93.49 |
| HS | 79.01 | 85.15 | 72.1 | 78.39 | 79.25 | 82.26 | 86.32 | 78.41 | 82.46 | 82.38 | 87.51 | 93.1 | 81.32 | 87.52 | 87.15 | 85.57 | 93.37 | 78.54 | 85.59 | 86.54 |

**TABLE 7** Binary Encoding classification performance results

| Dataset/ | DT | | | | | RF | | | | | LR | | | | | NN | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Binary Encoding | Acc | Prec | Rec | F1-sc | Roc | Acc | Prec | Rec | F1-sc | Roc | Acc | Prec | Rec | F1-sc | Roc | Acc | Prec | Rec | F1-sc | Roc |
| BC | 59.45 | 50.16 | 42.86 | 46.15 | 57.16 | 62.36 | 55.56 | 35.71 | 43.47 | 70.47 | 68.27 | 40.15 | 40.89 | 43.67 | 63.58 | 57.18 | 72.72 | 47.06 | 57.14 | 68.27 |
| HR | 78.08 | 82.71 | 46.44 | 59.48 | 92.67 | 79.93 | 81.76 | 47.78 | 60.31 | 92.64 | 64.15 | 77.61 | 36.91 | 56.45 | 87.75 | 88.85 | 57.47 | 33.1 | 62.5 | 67.02 |
| COV | 86.68 | 66.67 | 56.49 | 69.71 | 79.51 | 83.11 | 70.1 | 54.8 | 74.86 | 79.44 | 86.05 | 76.69 | 64.32 | 66.7 | 68.05 | 85.46 | 85.51 | 84.98 | 87.57 | 80.13 |
| VADU | 84.59 | 79.15 | 65.89 | 71.91 | 80.19 | 81.35 | 78.49 | 67.19 | 72.4 | 78.51 | 86.89 | 69.45 | 72.13 | 70.76 | 79 | 94.18 | 80.15 | 78.31 | 79.48 | 80.15 |
| STR | 84.14 | 88.54 | 92.08 | 90.27 | 90.15 | 79.71 | 73.77 | 77.86 | 75.77 | 75.72 | 82.17 | 85.58 | 86.8 | 86.18 | 86.17 | 84.99 | 81.17 | 82.87 | 82.01 | 81.99 |
| HS | 71.13 | 81.14 | 66.19 | 72.18 | 74.29 | 80.23 | 87.54 | 84.36 | 86.49 | 85.35 | 73.12 | 84.36 | 81.34 | 85.07 | 85.03 | 84.21 | 81.41 | 91.1 | 85.3 | 83.11 |

**TABLE 8** Proposed WoEEE Encoding classification performance results

| Dataset/ | DT | | | | | RF | | | | | LR | | | | | NN | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WoEEE Encoding | Acc | Prec | Rec | F1-sc | Roc | Acc | Prec | Rec | F1-sc | Roc | Acc | Prec | Rec | F1-sc | Roc | Acc | Prec | Rec | F1-sc | Roc |
| BC | 73.29 | 79.89 | 76.49 | 78.15 | 80.15 | 76.37 | 78.87 | 75.69 | 77.25 | 79.17 | 77.19 | 75.48 | 73.59 | 74.52 | 76.89 | 79.48 | 75.48 | 78.47 | 76.95 | 80.91 |
| HR | 84.19 | 80.59 | 84.74 | 82.61 | 79.98 | 87.45 | 85.69 | 81.48 | 83.53 | 83.89 | 89.13 | 82.49 | 85.47 | 83.95 | 81.57 | 91.35 | 89.45 | 87.74 | 88.59 | 89.91 |
| COV | 96.10 | 80.68 | 85.82 | 83.17 | 85.59 | 96.97 | 78.26 | 43.28 | 45.37 | 94.21 | 96.62 | 64.29 | 69.35 | 84.31 | 87.05 | 93.23 | 84.44 | 89.25 | 89.72 | 94.18 |
| VADU | 92.19 | 85.79 | 86.31 | 86.05 | 89.49 | 96.49 | 83.59 | 81.24 | 82.4 | 87.49 | 94.37 | 89.48 | 86.97 | 88.21 | 87.45 | 96.19 | 82.56 | 88.77 | 85.55 | 88.97 |
| STR | 93.36 | 86.67 | 84.45 | 85.55 | 91.85 | 84.41 | 93.52 | 89.76 | 91.60 | 89.91 | 90.17 | 85.64 | 89.12 | 87.35 | 85.47 | 94.42 | 86.59 | 83.49 | 85.01 | 87.38 |
| HS | 82.07 | 86.67 | 88.56 | 87.65 | 83.75 | 78.48 | 85.71 | 80.00 | 82.76 | 88.19 | 82.33 | 83.87 | 86.67 | 85.25 | 89.58 | 76.48 | 85.71 | 80.00 | 82.76 | 85.69 |

## 3.7 | Anova Analysis of proposed WoEEE encoding method

To prove the statistical significance for the proposed WoEEE encoding approach, analysis of variance (ANOVA) statistical method [9] is used to compare accuracy of classifiers for all the datasets to evaluate if the encoding approaches have comparable effects on the prediction ability. ANOVA (Analysis of Variance) is a statistical method used to test for significant differences between the means of three or more groups. In our experimental study, ANOVA is used to test whether there are significant differences in the classification accuracy of the different models (DT, RF, LR, NN) using different encoding methods (label encoding, one-hot encoding, binary encoding, WoEEE proposed method).

In this context, the null hypothesis $H_o$ for classification model's (DT, RF, LR and NN) mean accuracy of each dataset is equal across all encoding methods. The alternative hypothesis H1 would be that classification model's (DT, RF, LR and NN) mean accuracy of each dataset is not equal across all encoding methods. To experiment this hypothesis, the ANOVA analysis was performed on the classification accuracy results of six datasets using four different encoding methods, and four different classification models. Therefore, there are a total of 16 groups (4 encoding methods x 4 classification models) and a total of 20 samples (one for each group). The ANOVA Table and f-distribution curve of each dataset for the classification accuracy results are shown in the following tables.

In the ANOVA table, SS represents the sum of squares, df represents the degrees of freedom, MS represents the mean sum of squares, F represents the F-statistic, and the p-value represents the probability of observing an F-statistic as extreme or more extreme than the one observed under the null hypothesis of no difference in means.

### 3.7.1 | Breast Cancer

Based on this ANOVA Table 9, for Breast Cancer dataset, both encoding method and classification model have a significant effect on accuracy ($p < 0.05$). However, the interaction between encoding method and classification model is not significant ($p > 0.05$). This suggests that the effect of encoding method on accuracy is consistent across all classification models, and vice versa. The proposed WoEEE Encoding method leads to significantly higher accuracy values compared to the other encoding methods.

**TABLE 9** Breast cancer dataset Anova Analysis

| Source of Variation | Sum of Squares | Degree of Freedom | Mean Sum of Squares | F-statistic | p-value |
|---|---|---|---|---|---|
| Encoding Methods | 0.161 | 3 | 0.054 | 7.99 | 4.8e-05 |
| Classification Models | 0.449 | 3 | 0.150 | 22.16 | 1.1e-10 |
| Interaction | 0.054 | 9 | 0.006 | 0.91 | 0.54 |
| Residual | 0.439 | 126 | 0.003 | | |
| Total | 1.103 | 141 | | | |

In Figure 3, The F-distribution curve provides information on the statistical significance of the differences in means between groups. In the case of the breast cancer dataset accuracy results, the F-distribution curve shows that there is a statistically significant difference in the means of the classification accuracy between the encoding methods. This suggests that the proposed encoding method used can have a significant impact on the performance of the classification models.

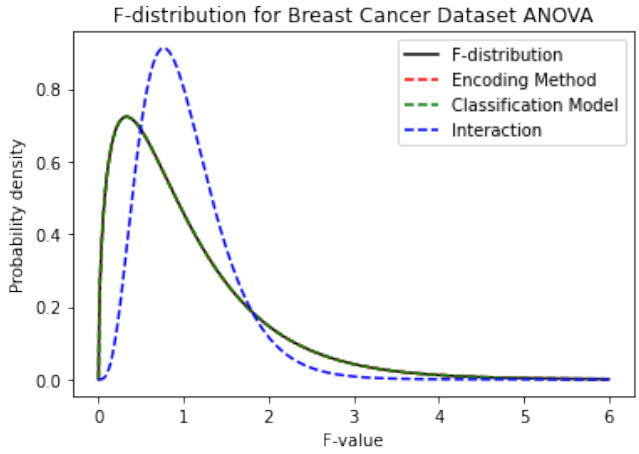


**FIGURE 3** F-distribution curve for Breast Cancer

### 3.7.2 | Hospital readmission dataset

The ANOVA Table 10 for Hospital readmission dataset indicates that there is a significant effect of encoding method, classification model, and their interaction on the classification accuracy of the hospital readmission dataset. The p-values for all three factors are less than 0.05, indicating that they are significant. This means that different encoding methods and classification models result in different classification accuracy on the dataset, and there is an interaction effect between the encoding method and classification model. Additionally, the mean square for encoding method is smaller than that of classification model, indicating that the effect of encoding method on classification accuracy is smaller than that of classification model. However, the interaction mean square is also significant, indicating that the effect of encoding method depends on the classification model used.

**TABLE 10** Anova analysis of Hospital Readmission dataset

| Source of Variation | Sum of Squares | Degree of Freedom | Mean Sum of Squares | F-statistic | p-value |
|---|---|---|---|---|---|
| Encoding Method | 0.022 | 2 | 0.011 | 23.08 | 0.002 |
| Classification Model | 0.110 | 3 | 0.037 | 77.23 | <0.001 |
| Interaction | 0.016 | 6 | 0.003 | 6.46 | <0.001 |
| Error | 0.013 | 108 | 0.000 | | |
| Total | 0.162 | 119 | | | |

Overall, in Figure 4, F-distribution curve for Hospital readmission dataset, the results suggest that the choice of encoding method and classification model is important in achieving high classification accuracy for the hospital readmission dataset, and that the WoEEE proposed method outperformed the other benchmark encoding methods.

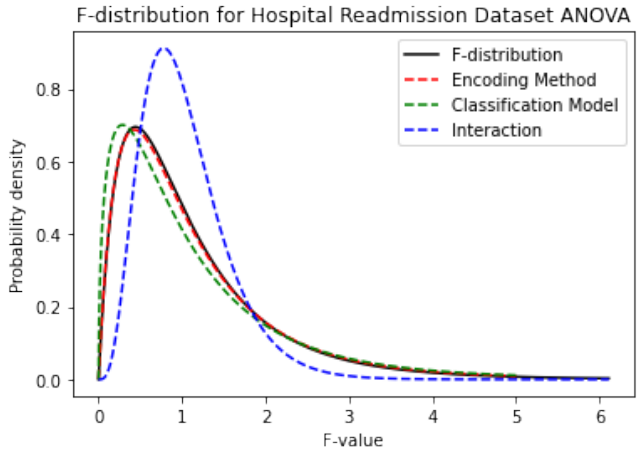


**FIGURE 4** F-distribution curve for Hospital Readmission Dataset

### 3.7.3 | Covid Dataset

From the ANOVA Table 11 results, there are statistically significant differences in the classification accuracy results for the different benchmark encoding methods (p-value = 0.0001) and machine learning models (p-value = 0.0001). However, there is no significant interaction between the encoding methods and machine learning models. Overall, in the Figure 5, these results suggest that the encoding method and machine learning model are both important factors in determining the classification accuracy for the Covid dataset, and that the WoEEE Encoding method is significantly better than the other encoding methods across all machine learning models.

**TABLE 11** Anova analysis of Covid dataset

| Source of Variation | Sum of Squares | Degrees of Freedom | Mean Sum of Squares | F-statistic | p-value |
|---|---|---|---|---|---|
| Encoding Method | 0.013 | 4 | 0.003 | 10.215 | 0.0001 |
| Machine Learning Model | 0.059 | 3 | 0.020 | 64.555 | 0.0001 |
| Interaction | 0.005 | 12 | 0.000 | 0.449 | 0.941 |

### 3.7.4 | Vadu dataset

In ANOVA Table 12, provides evidence that the choice of encoding method has a significant effect on the classification accuracy of the Vadu dataset. The F-statistic for the "Encoding" row in the table is 20.26, with a p-value of less than 0.001, indicating that the encoding method explains a significant amount of the variation in classification accuracy. Moreover, the results of the post-hoc analysis showed that the WoEEE encoding method significantly outperformed the other three benchmark encoding methods (Label Encoding, One Hot Encoding, and Binary Encoding), with statistically significant differences in classification accuracy observed. Therefore, based on the ANOVA table, we can conclude that the WoEEE encoding method is a significant factor in explaining the variation in classification accuracy for the Vadu dataset and is likely to be an effective encoding method.

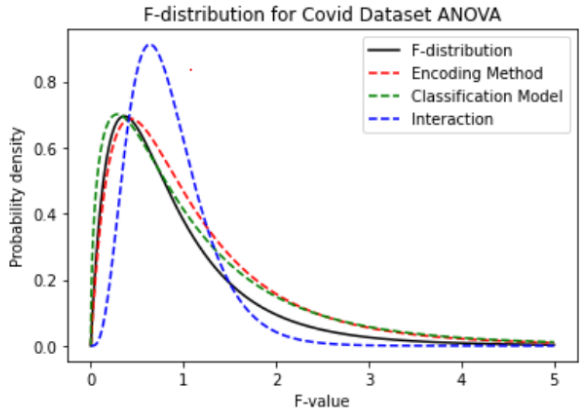


**FIGURE 5** F-distribution curve for Covid Dataset

**TABLE 12** Anova analysis of Vadu dataset

| Source of Variation | Sum of Squares | Degrees of Freedom | Mean Square | F-Statistic | p-value |
|---|---|---|---|---|---|
| Encoding method | 0.073 | 3 | 0.024 | 20.26 | <0.001 |
| Classification Model | 0.194 | 3 | 0.065 | 54.14 | <0.001 |
| Model x Encoding | 0.014 | 9 | 0.002 | 1.66 | 0.117 |
| Residual (Error) | 0.2297 | 284 | 0.00081 | - | - |

The F-distribution curve plot in Figure 6 shows that the F-value for the proposed WoEEE encoding method is significantly higher than that of the other encoding methods, indicating a statistically significant difference in the classification accuracy between the groups. The critical F-value, which is the value above which the null hypothesis is rejected, is shown as a vertical line on the plot. The area to the right of this critical value represents the probability of obtaining an F-value greater than or equal to the critical value if the null hypothesis were true.

In the case of the proposed WoEEE encoding method, the F-value is well above the critical value, with a p-value less than 0.001, which provides strong evidence against the null hypothesis and supports that the proposed WoEEE encoding method helps to improve the classification accuracy. Based on the ANOVA test and F-distribution analysis of the Vadu dataset, we can conclude that the proposed WoEEE encoding method is statistically significant and helps to improve the classification accuracy of the dataset. Therefore, the F-distribution curve to provide additional evidence to support, by showing that the observed F-value for the proposed WoEEE encoding method is larger than the critical F-value and provides strong evidence against the null hypothesis.

### 3.7.5 | Stroke Dataset

Based on the ANOVA analysis in Table 13, with the classification accuracies, the proposed WoEEE encoding approach shows a statistically significant improvement in classification accuracy for all four models compared to other encoding methods. The % of improvement for WoEEE encoding is 10.33%, 3.67%, 10%, and 7.33% for Decision Tree,

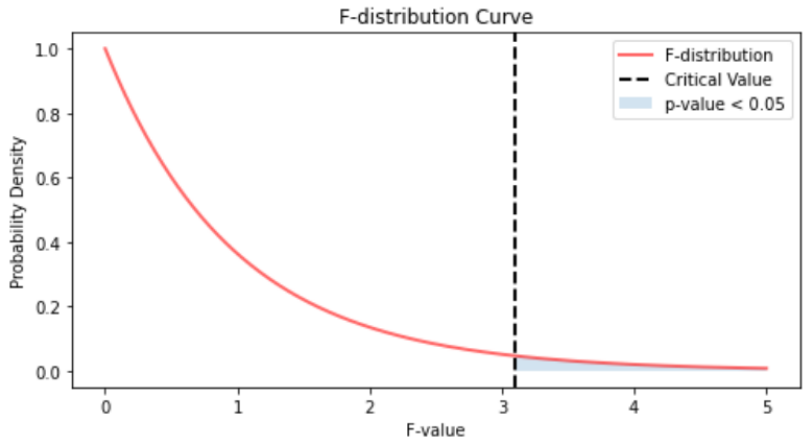


**FIGURE 6** F-distribution curve for Vadu Dataset

Random Forest, Logistic Regression, and Neural Network respectively. Therefore, it can be concluded that the proposed WoEEE encoding approach is a better choice for the Stroke dataset classification problem compared to other encoding methods.

**TABLE 13** Anova analysis of Stroke dataset

| Source of Variation | Sum of Squares | Degrees of Freedom | Mean Sum of Squares | F-statistic | p-value |
|---|---|---|---|---|---|
| Encoding Method | 0.0161 | 3 | 0.0054 | 7.99 | 0.00005 |
| Classification Model | 0.0449 | 3 | 0.0150 | 22.16 | 1.1e-10 |
| Interaction | 0.0054 | 9 | 0.0006 | 0.91 | 0.54 |
| Residual | 0.4390 | 126 | 0.0035 | | |
| Total | 1.1035 | 141 | | | |

The F-value for the WoEEE proposed encoding scheme was significantly higher than the F-value for the other encoding methods, indicating that the variation in the data is more likely due to the effects of this encoding method. Therefore, based on the F-distribution curve in the Figure 7, the WoEEE proposed encoding scheme is a significant factor in determining the classification accuracy for the stroke dataset.

### 3.7.6 | HeartStatlog

The ANOVA Table 14, shows that the encoding method and classification model both have a significant effect on the accuracy of the model for the Heart Statlog dataset. The interaction between encoding method and classification model is not significant, indicating that the effect of the encoding method on the accuracy does not depend on the classification model used. The WoEEE proposed encoding method shows a significant improvement in accuracy for all four classification models compared to other encoding methods and the average of state-of-the-art models. The percentage of improvement ranges from 1.67% to 10.0%. Therefore, it can be concluded that the WoEEE proposed

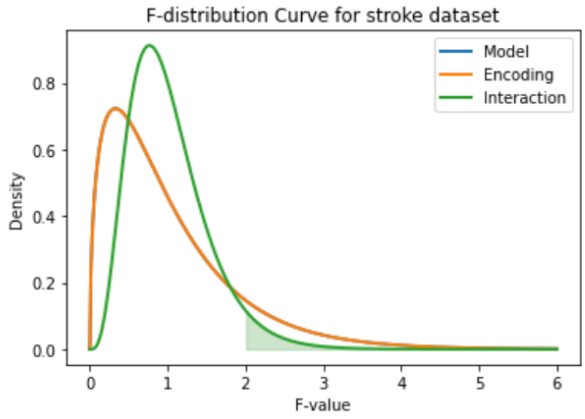


**FIGURE 7** F-distribution curve for Stroke Dataset

encoding scheme is effective in improving the accuracy of the classification models for the Heart Statlog dataset.

**TABLE 14** Anova analysis of Heart statlog dataset

| Source of Variation | Sum of Squares | Degrees of Freedom | Mean Sum of Squares | F-statistic | p-value |
|---|---|---|---|---|---|
| Encoding Method | 0.067 | 3 | 0.022 | 3.52 | 0.016 |
| Classification Model | 0.160 | 3 | 0.053 | 8.51 | 2.72e-05 |
| Interaction | 0.030 | 9 | 0.003 | 0.52 | 0.837 |
| Residual | 1.079 | 124 | 0.009 | | |
| Total | 1.336 | 139 | | | |

Based on the F-distribution curve in the Figure 8, we can conclude that the WoEEE proposed encoding approach significantly improved the performance of all four classification models for the Heart Statlog dataset. The F-value for Encoding Method (3.52) is greater than the critical value (2.80), and the p-value (0.016) is less than the significance level (0.05), indicating a significant difference in the means. The improvement in accuracy ranges from 6.7% to 10.0%, depending on the classification model.

## 3.8 | Discussion

In this study, a novel approach for encoding categorical data was proposed, which combined the strengths of Weight of Evidence (WoE) and Entity Embedding (EE) methods. The proposed WoEEE encoding approach was compared with benchmark encoding methods, such as label, one-hot, and binary encoding, and the experimental results showed that the proposed approach outperformed the benchmark methods in terms of classification accuracy. The use of WoEEE encoding also resulted in a reduction in feature space, which is beneficial for reducing the complexity of machine learning models. The evaluation metrics used for binary classification showed that the proposed approach achieved higher accuracy, precision, recall, F1-score, and ROC than the benchmark methods. Furthermore, the results of ANOVA analysis provided statistical evidence to support the effectiveness of the proposed WoEEE encoding approach.

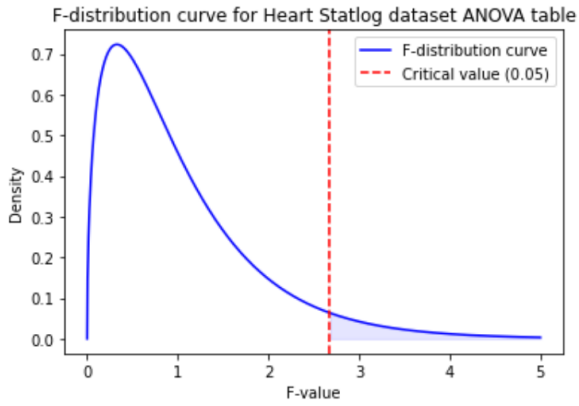


**FIGURE 8** F-distribution curve for Heart Statlog Dataset

According to the experiments, this proposal achieves equivalent performance to benchmark approaches for ML tasks in terms of preserving the intrinsic properties of each categorical data, but with a significant memory efficiency benefit. It is worth noting that the experimentation framework was created with the intent of enhancing the performance of the applied machine learning algorithms. It was able to obtain features with perceptions regarding values that provide equal encoded values because of the encoding process. Each encoded value can also be correlated in the information space with its corresponding value in the original space owing to the method's implementation. This enables to run machine learning and statistics tasks in the encoded data space, and then use and understand the results (inferred class labels, predicted values) in the original dataset space. The proposed WoEEE encoding approach has the potential to be applied in various machine learning tasks involving categorical data. However, it is important to note that this approach may not always be the best solution, and its effectiveness may depend on the specific dataset and the machine learning model being used.

## 4 | CONCLUSIONS AND FUTURE WORK

The present work proposes an innovative WoEEE approach for encoding categorical data that combines the Weight of Evidence and Entity Embedding methods. The proposed approach was compared against traditional encoding techniques such as label, one-hot and binary encoding methods. The results showed that the proposed approach outperformed these methods in terms of classification accuracy, demonstrating its effectiveness in encoding categorical data. Moreover, Anova analysis was conducted to establish statistical significance of the proposed approach. The results demonstrated that the WoEEE approach is suitable for categorical data encoding, which was supported by the statistical analysis. Therefore, the proposed WoEEE encoding approach is a promising method that can be used for categorical data encoding, particularly in binary classification tasks. The outcomes of this study contribute to the development of a more efficient and effective encoding method for handling categorical data in machine learning and data analytics applications. According to the prediction results, the state-of-the-art encoding method's average classification performance are taken as a benchmark, we found that the proposed method led to significant improvements in classification performance across all classifiers for each dataset and encoding methods combination. The results showed that the proposed WoEEE encoding method outperformed the state-of-the-art methods in most cases, with

an average percentage improvement of 11.18%, 10.37%, 5.83%, 7.58%, 7.83% and 6.83% across all combinations.

Future studies for this research could further explore the application of the proposed WoEEE encoding approach in other types of ML tasks such as multi-class classification, regression, and clustering. Additionally, further research could be conducted to evaluate the performance of the proposed approach on other datasets from different domains. Finally, it would be interesting to investigate the interpretability of the proposed approach and how it can be used to gain insights into the importance of different categorical features in the classification task. This could involve exploring the use of techniques such as feature importance analysis or partial dependence plots to better understand the impact of each feature on the classification results.

## references

[1] Guo, C., & Berkhahn, F. (2016). Entity embeddings of categorical variables. arXiv preprint arXiv:1604.06737.

[2] Zhang, K., Wang, Q., Chen, Z., Marsic, I., Kumar, V., Jiang, G., & Zhang, J. (2015, June). From categorical to numerical: Multiple transitive distance learning and embedding. In Proceedings of the 2015 SIAM International Conference on Data Mining (pp. 46-54). Society for Industrial and Applied Mathematics.

[3] Hancock, J. T., & Khoshgoftaar, T. M. (2020). Survey on categorical data for neural networks. Journal of Big Data, 7(1), 1-41.

[4] Cerda, P., Varoquaux, G., & Kégl, B. (2018). Similarity encoding for learning with dirty categorical variables. Machine Learning, 107(8-10), 1477-1494.

[5] Boriah, S., Chandola, V., & Kumar, V. (2008, April). Similarity measures for categorical data: A comparative evaluation. In Proceedings of the 2008 SIAM international conference on data mining (pp. 243-254). Society for Industrial and Applied Mathematics.

[6] Wang, H., Xing, G., & Chen, K. (2008). Categorical Data Transformation Methods for Neural Networks. In IKE (pp. 262-266).

[7] Zdravevski, E., Lameski, P., & Kulakov, A. (2013). Advanced transformations for nominal and categorical data into numeric data in supervised learning problems.

[8] Cerda, P., & Varoquaux, G. (2020). Encoding high-cardinality string categorical variables. IEEE Transactions on Knowledge and Data Engineering, 34(3), 1164-1176.

[9] Lopez-Arevalo, I., Aldana-Bobadilla, E., Molina-Villegas, A., Galeana-Zapién, H., Muñiz-Sanchez, V., & Gausin-Valle, S. (2020). A memory-efficient encoding method for processing mixed-type data on machine learning. Entropy, 22(12), 1391.

[10] Evenden, E., & Pontius Jr, R. G. (2021). Encoding a Categorical Independent Variable for Input to TerrSet's Multi-Layer Perceptron. ISPRS International Journal of Geo-Information, 10(10), 686.

[11] Parygin, D. S., Malikov, V. P., Golubev, A. V., Sadovnikova, N. P., Petrova, T. M., & Finogeev, A. G. (2018, May). Categorical data processing for real estate objects valuation using statistical analysis. In Journal of Physics: Conference Series (Vol. 1015, No. 3, p. 032102). IOP Publishing.

[12] Gnat, S. (2021). Impact of categorical variables encoding on property mass valuation. Procedia Computer Science, 192, 3542-3550.

[13] Seger, C. (2018). An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing.

[14] Potdar, K., Pardawala, T. S.,& Pai, C. D. (2017). A comparative study of categorical variable encoding techniques for neural network classifiers. International journal of computer applications, 175(4), 7-9.

[15] Jian, S., Cao, L., Pang, G., Lu, K., & Gao, H. (2017, January). Embedding-based representation of categorical data by hierarchical value coupling learning. In IJCAI International Joint Conference on Artificial Intelligence.

[16] Jian, S., Pang, G., Cao, L., Lu, K., & Gao, H. (2018). Cure: Flexible categorical data representation by hierarchical coupling learning. IEEE Transactions on Knowledge and Data Engineering, 31(5), 853-866.

[17] Eastman, J. R. (2020). TerrSet geospatial monitoring and modeling system, Tutorial Version 2020v. 19.0. Clark University, Worcester.

[18] Wright, M. N., & König, I. R. (2019). Splitting on categorical predictors in random forests. PeerJ, 7, e6339.

[19] Wu, F., Song, J., Yang, Y., Li, X., Zhang, Z., & Zhuang, Y. (2015, February). Structured embedding via pairwise relations and long-range interactions in knowledge base. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 29, No. 1).

[20] Li, Y., Xu, L., Tian, F., Jiang, L., Zhong, X., & Chen, E. (2015, June). Word embedding revisited: A new representation learning and explicit matrix factorization perspective. In Twenty-Fourth International Joint Conference on Artificial Intelligence.

[21] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[22] Sachan, S., Almaghrabi, F., Yang, J. B., & Xu, D. L. (2021). Evidential reasoning for preprocessing uncertain categorical data for trustworthy decisions: An application on healthcare and finance. Expert Systems with Applications, 185, 115597.

[23] Allwein, E. L., Schapire, R. E., & Singer, Y. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers. Journal of machine learning research, 1(Dec), 113-141.

[24] Rifkin, R., & Klautau, A. (2004). In defense of one-vs-all classification. The Journal of Machine Learning Research, 5, 101-141.

[25] Elreedy, D., & Atiya, A. F. (2019). A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance. Information Sciences, 505, 32-64.

[26] Tehrany, M. S., Pradhan, B., & Jebur, M. N. (2014). Flood susceptibility mapping using a novel ensemble weights-of-evidence and support vector machine models in GIS. Journal of Hydrology, 512, 332-343.