IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 June 2022, accepted 23 June 2022, date of publication 29 June 2022, date of current version 8 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3187287

RESEARCH ARTICLE

The Categorical Data Conundrum: Heuristics for Classification Problems–A Case Study on Domestic Fire Injuries

DENIS REILLY, MARK TAYLOR, PAUL FERGUS, CARL CHALMERS, AND STEVEN THOMPSON

School of Computer Science and Mathematics, Liverpool John Moores University, Liverpool L3 3AF, U.K.

Corresponding author: Denis Reilly (d.reilly@ljmu.ac.uk)

ABSTRACT Machine learning is well developed amongst the scientific community in terms of theoretical foundations (statistics and algorithms) and frameworks (Tensorflow, PyTorch, H2O). However, machine learning is heavily focused on numerical data, or numerical data mixed with some categorical data. For numerical datasets, scientists and engineers can enjoy reasonable success with only a limited knowledge of theoretical foundations and the inner workings of machine learning frameworks. However, it is a different story when dealing with purely categorical datasets, which require a deeper understanding of machine learning frameworks and associated encodings and algorithms in order to achieve success. This paper addresses the issues in handling purely categorical datasets for multi-classification problems and provides a set of heuristics for dealing with purely categorical data. In particular, issues such as pre-processing, feature encoding and algorithm selection are considered. The heuristics are then demonstrated through a case study, based on a categorical data set of domestic fire injuries, covering a 10-year period. Novel contributions are made through the heuristics and the performance analysis of different encoding techniques. The case study itself also makes a novel contribution through the classification of different types of injuries, based on related features.

INDEX TERMS Categorical data, machine learning, feature encoding, algorithms, classification, fire injuries.

I. INTRODUCTION

The British statistician Karl Pearson was responsible for some of the earliest work in the 1900s, which considered categorical data [1]. The subsequent development of methods for dealing with categorical data stemmed from studies in the social and biomedical sciences. For example, politics is often measured as: leftist, liberal, moderate, or conservative. Diagnosis in relation to breast cancer typically uses categories of: normal, benign, probably benign, or suspicious and malignant.

What with the growth in machine learning over recent decades, many datasets are likely to contain categorical variables, some moreso than others. Datasets used in medicine for the treatment of diseases such as cancer and pain relief are rich in terms of categorical variables [2]. Other datasets may represent opinions over controversial issues on social media, which require sentiment analysis and natural language processing to extract useful information [3].

Datasets involving numeric variables (whether continuous or discrete) tend to be easier to deal with and modern machine learning frameworks (Tensorflow, PyTorch, H2O) can be applied to these datasets with relative ease. Numeric variables are more easily interpreted and lend themselves to processes such as forecasting and prediction (e.g. weather data). In contrast, categorical variables tend to hide (even mask) a great deal of the interesting information in a dataset [4]–[6]. It is not so easy to see trends and make predictions or forecasts when categorical variables dominate the dataset [7], [8]. This makes it crucial to develop systematic methods and heuristics

The associate editor coordinating the review of this manuscript and approving it for publication was Mehul S. Raval.

for dealing with such variables. Without such methods, vital information may be missed, which may make the difference between a patient surviving cancer, as opposed to them dying from cancer.

A. HEURISTIC GUIDELINES FOR CATEGORICAL DATA

The foundations of machine learning lie in classical statistics and as such machine learning algorithms operate on numerical data. Consequently, the first hurdle in dealing with categorical variables is conversion (or encoding) into a numerical format. However, this is not the only issue as categorical variables require closer inspection of the dataset itself. Some values may have to be dropped to avoid dimensionality overload. Effects of the 'Other' category also need to be accounted for when examining accuracy – numerical data does not suffer from the 'Other' category.

This paper first examines the application of machine learning to categorical datasets and goes on to develop a set of heuristics to guide the decision making process when dealing with datasets with a significant categorical variable content. The approach favours heuristics over algorithms in accordance with [9] in that an algorithm provides rules to yield an exact, reliable result that works for every case. Heuristics provide an incomplete set of suggestive 'rules of thumb' that work in some cases, but not in all.

The use of heuristics is reinforced in [10], which considers how heuristics can be used to find non-optimal, but good acceptable solutions for intractable problems. In other words, the guarantee of finding optimal solutions is sacrificed for the sake of achieving good solutions in a limited time. A heuristic is an informal or approximate algorithm that may not explore all possible states of the problem, but will typically explore the most likely states. [11].

Heuristics are often applied in games. For example when developing an algorithm for a chess game one would need to consider every possible move at some depth level and apply an evaluation function to the board state. A heuristic would exclude full branches that begin with obviously bad moves. Another heuristic would be to always take the opponent's queen, if possible.

Many researchers have noticed that heuristic procedures (e.g. sampling many special cases by trial and error) often lead to greater insights and stimulus by engaging thought processes [12]. In contrast, users tend to rely on algorithms 'doing all of the work' without questioning the outcome, often to their detriment. In short, heuristics are better at engaging the user, whereas algorithms are treated as black boxes for which the outcomes are taken for granted.

B. CATEGORICAL DATA ISSUES

Advancements in various machine learning frameworks accommodate complex categorical data types like text labels. Typically, feature engineering involves some form of transformation of these categorical values into numeric labels, followed by the application of an encoding technique on these values. Various encoding techniques exist and the choice of the appropriate technique can be crucial for machine learning algorithms and hence the feature engineering process.

Categorical data can also suffer from the "dirty data problem" [13], which can occur during the collection of the dataset. Typical issues include:

- Capitalization (e.g. data includes both 'liverpool' and 'Liverpool').
- Extraneous data (e.g. name and title, instead of just the name).
- Abbreviations (e.g. Dr for Doctor).
- Encoding formats (e.g. ASCII, EBCDIC, etc.).
- Special characters (space, colon, hyphen, parenthesis e.g. 'fire-service' and 'fire service').

C. CATEGORICAL DATA CLASSIFICATION

Machine learning is broadly classified into supervised and unsupervised learning. Regression and classification are the two main tasks conducted for supervised learning (outcomes for all feature points are mapped). Clustering is the main unsupervised learning task (outcomes are not given for the data points). Typically, regression is applied to continuous or discrete numerical data, whereas classification is often applied to categorical data. Classification problems may either be binary (yes/no, true/false) or multi-classification.

The prime concern of this paper is classification problems (more specifically multi-classification), although some of the findings may also be applied to clustering problems. In particular, the heuristics developed from the research are applied to a dataset of domestic fire injuries. The dataset, which is rich in categorical data, records various fire-related injuries as the target label and the associated features, including: age, gender, type of dwelling and circumstances of the injury (e.g. attempt to extinguish the fire, or discovery of the fire).

D. CONTRIBUTION TO KNOWLEDGE

The paper makes several novel contributions; the first being the heuristics themselves. The second is the performance analysis of different encoding techniques. The third is a case study, which demonstrates the application of the heuristics and shows how useful information can be extracted from a solely categorical data set relating to injuries suffered during domestic fires. Examination of the literature revealed little in terms of research to directly examine domestic fire injuries, or circumstances pertaining to those injuries. This further reinforces the need for the research presented herein.

II. RELATED WORK

Related work is reviewed on two fronts: first work that considers the challenges facing the use of machine learning techniques with categorical datasets and second, work that considers the classification and risk assessment of fire-related incidents.

A. MACHINE LEARNING APPROACHES TO CATEGORICAL DATASETS

Reference [14] provides a useful survey on categorical data for neural networks. The survey describes the main encoding and algorithmic techniques specific to categorical data. In particular, the survey considers big data sets, which render some encoding techniques impractical due to their running time complexity. Reference [15] provides a more succinct and specific evaluation of encodings for neural networks, based on the car evaluation dataset from the UCI Machine Learning Repository.

Reference [13] considers the challenges of "dirty" non-curated data, such as 'Pfizer International LLC', 'Pfizer Limited', and 'Pfizer Korea'. The approach shows that with high-cardinality categorical variables, one-hot encoding can become impracticable due to the resulting high-dimensional feature matrix. This shortcoming is addressed through a softer version of one-hot encoding, based on string similarity measures. The solution is presented as similarity encoding, which encodes the morphological resemblance between categories. This in turn leads to dimensionality reduction, which decreases the runtime of the learning process.

Useful heuristics are provided in [16], which describes ten tips or checks for machine learning problems in computational biology. The tips begin with techniques for pre-processing and cleaning the dataset and move on to consider guidelines for selecting the best algorithm – starting simple and working towards the more advanced. The tips themselves become increasingly more advanced, considering issues such as hyper-parameter tuning and techniques to minimize overfitting.

Feature selection is considered in [17], especially for datasets with many variables and features. The intention is to illustrate how the elimination of unimportant variables improves the accuracy and performance of classification. The work considers datasets with mixed numerical and categorical features and the conclusion is that Random Forest is seen as the most versatile algorithm that can handle a higher number of categorical variables and provide high levels of performance.

B. MACHINE LEARNING APPLIED TO FIRE DATASETS

There is a large body of work concerned with the use of machine learning techniques to predict forest fires, typified by [18]–[24]. These works use a variety of neural networks, decisions trees and random forest algorithms. There is also a body of work that applies ML to study emergency evacuation of buildings (including fire emergencies), typified by [25]–[27].

Other more focused research applies ML algorithms for the classification of occupational accidents (including fire accidents) [28]. This work uses a combination of Support Vector Machines (SVM), Neural Networks (NN) and Decisions Trees (DT) (C4.5 and C5.0 algorithms) for classification from a categorical dataset. Results from the SVM and NN algorithms are compared to determine the best classifier. The intermediate results are then passed to the DT algorithms for rule extraction.

Fire risk is considered in [29] which is concerned with the collection of datasets, based on property inspections, and the development of a predictive model for a large number of commercial buildings. A fire risk assessment scoring system is developed by [30], based on SVM. In [31] the use of Bayesian Networks (BN) is assessed to improve current fire risk analysis methods. The study found that BNs have significant advantages over existing tools currently used in fire safety engineering, such as fault trees and event trees. A similar theme emerges in [32] which considers how the use of NN for fire prediction in property provides advantages over existing methods.

Bayesian Networks (BN) are also applied to dwelling fires in [33]. In particular, a three-part BN model is developed to study dwelling fires and improve confidence in dwelling fire safety assessment. Case studies demonstrate how the model functions and provide evidence of its use for planning and accident investigation.

III. PRELIMINARIES

In a typical machine learning problem, many of the variables may be categorical, sometimes even all of them. Categorical variables can be nominal (e.g. Jewish, Hindu, Sikh, Muslim, Christian, Orthodox) or ordinal (e.g. extra-small, small, medium, large, extra-large), where the latter implies some form of ordering and the former does not. For categorical datasets, the application of the appropriate encoding techniques and choice of algorithm can have a significant impact on the model's accuracy, performance and prediction.

As mentioned previously, classification tasks involve supervised learning when a labelled dataset is available and the output/target variable is already known. Classification tasks are either binary classification (yes/no, true/false, allow loan/don't allow loan) or multi-classification i.e. classification into one of several target variable values (extra-small, small, medium, large, extra-large).

In machine learning, a label is the value to predict i.e. the y variable. A feature is an input variable and typically there will be many such features, possibly hundreds (or even thousands), specified as $\mathbf{X}$. The label and features are related as $y = f(\mathbf{X})$, where f is the function to be learned. There are several popular classification algorithms that can be used for categorical data and these are considered further below.

A. CLASSIFICATION ALGORITHMS

The main classification algorithms are: K-Nearest Neighbour (KNN), Naive Bayes (NB), Decision Tree (DT), Random Forest (RF), Artificial Neural Network (ANN) and Support Vector Machine (SVM).

1) K-NEAREST NEIGHBOUR (KNN)

KNN is a distance-based classifier, which implicitly assumes that the smaller the distance between two points (i.e. observations), the more similar they are. Distance (or dissimilarity) metrics are used to compute pairwise differences between observations. For continuous variables, the most common distance measures are the Euclidean, Manhattan, or Minkowski [34]. For categorical variables, the Hamming distance is used (1).

$$\begin{aligned} D_H &= \sum_{i=1}^{K} |x_i - y_i| \\ x &= y \Rightarrow D = 0 \\ x &\neq y \Rightarrow D = 1 \end{aligned} \tag{1}$$

2) NAÏVE BAYES (NB)

NB provides a simple, but effective classifier. Naïve Bayes is often applied to a dataset containing multiple features and an output label assuming one of two discrete values (i.e. binary classification). NB assumes independence between features i.e. the presence of a particular feature in a class is unrelated to the presence of any other feature. NB is suitable for very large datasets and often outperforms the more complex classification methods. If y is the class variable and X represents the parameters/features, Naïve Bayes can be expressed as (2).

$$P(y, X) \propto P(y) \prod_{i=1}^{K} P(x_i|y) \tag{2}$$

If the classification problem is multivariate, the class variable y with maximum probability can be determined using the *argmax* function (3).

$$P(y, X) = argmax_y \left[P(y) \prod_{i=1}^{K} P(x_i|y) \right] \tag{3}$$

3) DECISION TREE (DT)

DT is based on a tree representation in which each leaf node corresponds to a class label and attributes are represented on the internal nodes of the tree. DT serves as the basis for other tree-based algorithms (e.g. Random Forest), which provide popular non-parametric supervised learning algorithms for classification.

A decision tree with K leaves divides the feature space into K regions. The prediction function of a tree is then defined as (4).

$$\hat{y} = \hat{f}(x) = \sum_{i=1}^{K} c_i I\{x \epsilon R_i\} \tag{4}$$

$I\{x \epsilon R_i\}$ is the identity function, which returns 1 if x is in the subset R_n and 0 otherwise. K is the number of leaves in the tree, $R_i (1 \leq i \leq K)$ is a region in the feature space (corresponding to leaf i) and c_i is a constant (corresponding to region i) whose value is determined in the training phase of the algorithm.

For classification trees, two metrics are used to determine how to split a tree at a specific node. The Gini impurity measure (5) and the Entropy measure (6) are the criteria used for calculating Information Gain. DT algorithms use Information Gain to decide on which feature to split on at each step in the construction of the tree. The DT algorithm always tries to maximize Information Gain.

The Gini impurity measure is the probability of a random sample being classified incorrectly.

$$Gini = 1 - \sum_{i=1}^{K} p_i^2 \tag{5}$$

Entropy is an information theory metric that measures the impurity or uncertainty in a group of observations.

$$E = -\sum_{i=1}^{K} p_i \log_2 p_i \tag{6}$$

Conceptually, DTs are useful as they pose a series of questions. The answers at each stage lead to further questions in the series. These decisions and questions continue until a terminal node is reached after which further questions are not possible.

4) RANDOM FOREST (RF)

RF is based on DT and consists of a collection of tree-structured classifiers $\{h(\mathbf{X}, \Theta_i)\ i = 1, .., K\}$, where $\{\Theta_i\}$ are independently identically distributed random vectors. Each tree casts a unit vote for the most popular class in $\mathbf{X}$ [35].

A margin function is defined as (7).

$$mg(\mathbf{X}, Y) = av_i I(h_i(\mathbf{X}) = Y) - \max_{j \neq Y} av_i I(h_i(\mathbf{X}) = j) \tag{7}$$

where $I(h_i(\mathbf{X}) = Y)$ is the indicator function and av_i is the average number of votes at X, Y for the corresponding class. The margin function measures the extent to which the average number of votes at X,Y, for a specific class (i), exceeds the average number of votes for any other class.

In RF, multiple trees are grown and each tree gives a classification, based on the votes for that class. RF chooses the classification having the most votes (over all the trees in the forest).

A particularly useful aspect of RF for classification is that of feature importance. Several measures of feature importance have been proposed for RFs and these are discussed in [36].

5) ARTIFICIAL NEURAL NETWORK (ANN)

ANN can be shallow neural networks or deep neural networks. ANNs are often seen as the most powerful and popular class of machine learning algorithms as they can be used in many different problem domains. ANNs are based on artificial neurons and the underlying philosophy is that they learn from their own errors. Each neuron effectively calculates a "weighted sum" of its input, adds a bias and decides whether or not to fire. This can be expressed as a function f applied to a linear classifier $\mathbf{w}^{\mathrm{T}}\mathbf{x}+b$ (8). The decision function f could be a non-linear threshold function, a non-linear distance function or a probability-like sigmoid function.

$$y = f(\mathbf{w}^{\mathrm{T}}\mathbf{x}+b) \tag{8}$$

where **x** is the input features vector, **w** is the weights vector, which provides the connections between neurons that carry values. The higher the value, the larger the weight, and the greater the importance of a specific neuron on the input side of the weight. The bias value (b) shifts the activation function by adding a constant value, similar to a constant in any linear function.

A loss function is used to determine the difference between the actual values (y_i) and the predicted values ($\hat{y}_i$). Generally, mean squared error is used for regression problems and cross-entropy for classification problems. For multiclass problems, Categorical Cross Entropy (CCE) (9) is commonly used, where K is the number of classes and N is the number of observations.

$$CCE = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{K}\left[y_i^{(n)} log\left(\hat{y}_i^{(n)}\right) + \left(1-y_i^{(n)}\right)\log(1-\hat{y}_i^{(n)})\right] \qquad (9)$$

6) SUPPORT VECTOR MACHINE (SVM)

SVM is often applied to a dataset containing a very large number (infinite) of features, although in practice feature engineering may be used to reduce the number of features [37]. SVM is a supervised machine learning algorithm that can be used both for classification and regression analysis, but is more widely used for classification.

The SVM algorithm plots each data item as a point in K-dimensional space (where K is the number of features). The value of each feature is the value of a particular coordinate point and classification is performed by finding the hyper-plane that differentiates the two classes. As a classification algorithm SVM is suitable for categorical data. The 'support vectors' are data points that are the closest to the decision surface (or hyperplane), thereby rendering these data points the most difficult to classify.

Generally, input and output for SVMs are the same as for neural networks, and the characteristic equation is that of a linear classifier ($\mathbf{w}^{\mathrm{T}}\mathbf{x}+b$). The important differences are the maximized margin and hyperplane. The hypothesis function h is expressed as (10).

$$h(\mathbf{x}) = \begin{cases} +1 \; if \;\; \mathbf{w}\cdot\mathbf{x}+b \geq 0 \\ -1 \; if \;\; \mathbf{w}\cdot\mathbf{x}+b < 0 \end{cases} \qquad (10)$$

Through $h(\mathbf{x})$ points on or above the hyperplane are class $+1$ and points below are class -1.

With an ANN perceptron, the hyperplane is effectively found by iteratively updating the weights and minimizing the cost function. In contrast, SVM works by finding the optimal hyperplane which best separates the data and this is generally found by minimizing the Lagrange function L (11). Where α is the Lagrange multiplier.

$$\min L = 1/2\left\|\mathbf{w}\right\|^2 - \sum_{i=1}^{K}\alpha_i\left[y_i\left(\mathbf{w}\cdot\mathbf{x}_i+b\right)-1\right] \qquad (11)$$

B. DATA ENCODING

Categorical variables must be converted into numerical values before they can be processed by machine learning algorithms. The performance of various machine learning algorithms and the subsequent results vary depending upon the encoding technique used.

Encoding techniques can be broadly classified into three main categories:

1. Classic encoders – which are simple and straightforward to use. Examples include Label, Ordinal and One-hot encoding.
2. Bayesian encoders – which make use of information from a dependent variable as well as the categorical variable itself. Bayesian encoders output just one column, which eliminates high-dimensionality issues present in other encoders. Examples include Target, Leave-one-out and James-Stein encoding.
3. Contrast encoders – which work by typically comparing the mean of the independent variable and the dependent variable over their levels. Examples include Helmert, Backward Difference and Polynomial encoding.

The most commonly used encoding techniques are introduced briefly below and elaborated further in section IV. A more detailed consideration of categorical encoding techniques is provided in [38].

1) LABEL ENCODING

Label encoding (also referred to as Integer encoding) is used for nominal variables, where the variable consists of a finite set of discrete classes with no relationship existing between the classes. Label encoding assigns each category a value from 1 to N (where N is the cardinality of the feature). A major issue with this encoding is that the algorithm will impose an ordering even though no such order exists between the categories.

2) ORDINAL ENCODING

Ordinal encoding is used for ordinal variables, which consist of a finite set of discrete classes with a ranked ordering between the classes. The integer values have natural ordered relationships between each other and a machine learning algorithm will utilize these relationships.

3) ONE-HOT ENCODING

One-hot encoding is often used for nominal categorical variables for which no ordinal relationship exists and label (integer) encoding is insufficient.

For a random categorical variable X with n distinct values $x_1, x_2, \ldots x_n$. The one-hot encoding of a particular value x_i is a vector v in which every component is zero except for the *ith* component, which has the value 1.

For example, if X takes values from the set $S = a, b, c$ and $x_1 = a, x_2 = b, x_3 = c$. A One-hot encoding for x is *(1,0,0)*, *(0,1,0)*, and *(0,0,1)*.

4) DUMMY VARIABLE ENCODING

Dummy variable encoding is essentially a compact form of one-hot encoding, which eliminates redundancy. For the example above, if *(1, 0, 0)* is *a* and *(0, 1, 0)* is *b* another binary variable is not needed for *c*. Dummy variable encoding represents *C* categories with *C-1* binary variables.

5) HELMERT ENCODING

Helmert encoding is a widely used contrast encoder. In Helmert encoding, the mean of the dependent variable for a level is compared to the mean of the dependent variable over all subsequent levels. In Reverse Helmert Coding, the comparison is performed over all previous levels.

6) TARGET ENCODING (OR MEAN ENCODING)

Target encoding is a Bayesian encoding technique in which the mean of the target variable is calculated for each category and the category variable is then replaced with the mean value. For the categorical target variables (labels), the posterior probability of the target is used to replace each category. Target encoding is often used for important features.

7) JAMES-STEIN ENCODING

James-Stein encoding is a Bayesian encoding technique, which uses the mean target value for the observed feature and the mean target value to obtain a weighted average. James-Stein encoding is best suited to features with normal distributions.

8) HASH ENCODING

Hash encoding is suited to categorical variables with high cardinality. The encoding relies on a *hash function*, which maps the value of a category to an integer. This effectively converts categorical variables to a higher dimensional space of integers and the distance between two vectors of categorical variables is approximately maintained.

As considered in section IV, the choice of encoding can be significant. Ordinal variables will generally take care of themselves by way of ordinal encoding. One-hot encoding is a popular choice for nominal variables, but care is needed to avoid an explosion in the number of variables. Furthermore, one-hot encoding should be avoided when tree-based algorithms are used. Section 3 provides a flowchart to guide the choice of the encoding technique.

IV. HEURISTICS FOR CATEGORICAL DATA ANALYSIS

This section describes the heuristics for preprocessing, choice of encoding and algorithm selection when dealing with multi-classification of categorical datasets.

A. PREPROCESSING

Preprocessing of the dataset is often overlooked as researchers believe that the machine learning algorithm will "work its magic" and "paper over" noise or inconsistencies within the dataset. However, the effects of noise and inconsistencies are amplified when dealing with categorical datasets. The following pre-processing steps are advised to avoid error amplification.

Preprocessing Steps:

A: Clean data to remove dirty data and make categorical values consistent - this may appear simplistic, but it is invaluable for improving the accuracy of the model and hence the results. Ideally, an effective data collection protocol would prevent any dirty data, but inevitably some degree of dirty data is usually present. The most common data cleaning operations are listed below. Several of these can be performed using in-built procedures in spreadsheets like Excel.

  i. Remove duplicates - duplicates will skew data and affect results.
  ii. Remove irrelevant data - this can slow performance and confuse the analysis. Data such as date of birth, URLs, HTML tags can often be removed.
  iii. Standardize capitalization – otherwise, new erroneous categories will be introduced.
  iv. Remove formatting – if data is gathered from different document formats.
  v. Fix errors – errors such as spelling mistakes can be corrected using a spell-checker to avoid erroneous categories and inaccurate results.
  vi. Fix missing values – either remove the observation with the missing value, or enter the missing data (if applicable).

B. If 'Age' is a feature choose an appropriate representation. 'Age' is technically continuous and ratio, but from a machine learning perspective, Age is generally treated as either nominal or ordinal depending on the situation. Typically age may be split into categories with bucket sizes of 10 years i.e. 0-9, 10-19, 20-29,…, 80-89, 90-99. However, in many instances the first and last categories may not represent many cases as young children or people approaching 100 may not feature in the data. Often the final category/bucket may be represented as '80+'.

C. Obtain frequency summaries to identify important and less important features. Frequency summaries are simply counts of the number of occurrences for each category of each variable. They are useful for identifying values that are of the greatest importance and also those of lesser importance.

D. Based on frequency counts, examine features to see if any can be consolidated to reduce cardinality - less frequent values can often be sacrificed and put into an 'Other' category without loss of accuracy.

B. CHOICE OF ENCODING

The main encoding techniques used for categorical data were introduced in section III. If the dataset has categorical features with high cardinality there may be certain problems due to overfitting and data leakage. If a one-hot encoder is used, the dataset can suddenly become very wide and sparse, which

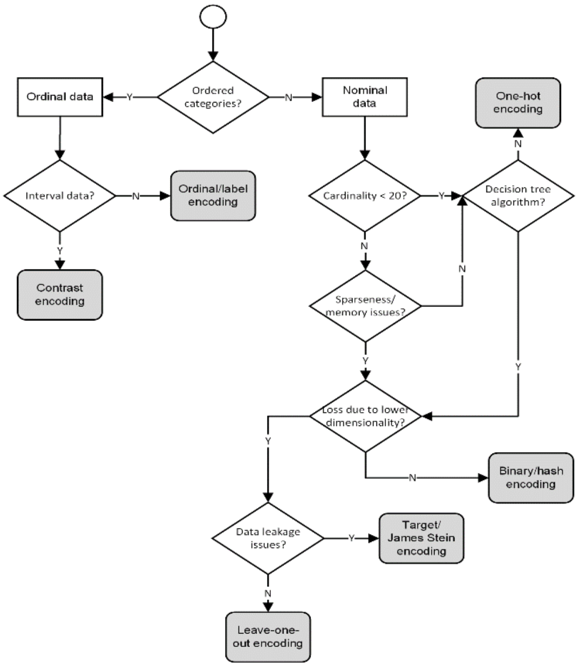


FIGURE 1. **Flowchart for selection of categorical encoding technique.**

in turn can present computational challenges. If tree-based algorithms are to be used the trees may grow in one direction, which may lead to overfitting. These issues (and others need to be considered when choosing the encoding technique. Experience from this research revealed that the issues can either be addressed conceptually or technically.

From a conceptual point of view, the dataset can be examined to see if any values can be consolidated to reduce the cardinality, as described in IV. *A Preprocessing* above. The technical solution involves choosing the appropriate encoding technique, based on three criteria:

- The characteristics of the dataset.
- The algorithm to be used (e.g. tree-based vs linear).
- The accuracy required.

The flowchart of Fig. 1 provides a guide to choosing an encoding technique, based on the above criteria. Cardinality < 20 effectively sets a threshold value after which one-hot encoding becomes impractical. One-hot encoding uses a binary representation and if the number of categories for a variable is around 20 the size of the binary number representations proves to be excessive.

C. CHOICE OF ALGORITHM

Previous research has compared the application and performance of classification algorithms, KNN, NB, RF, ANN and SVM, as typified by [39]–[41]. Additionally, certain algorithms are more suited to certain applications. For example, for image classification, ANN will offer the best performance. For Natural Language Processing (NLP), SVM will be superior. RF is often used in business and financial classification problems, such as fraud detection, loan defaulting and customer loyalty. However, comparisons in the literature tend to favour benchmark numerical datasets, whereas the concern of this paper is largely categorical datasets.

Given the lack of categorical benchmark datasets, the systematic choice of a classification algorithm should be based on sound criteria. Experience from this research revealed that such criteria can be grouped under the requirements of the classification problem and the dataset itself, as below.

Requirements:

- Accuracy
- Performance/speed
- Training
- Scalability
- Parallelization
- Extrapolation/interpolation
- Amount of parameter tuning
- Parametric/non-parametric

Dataset:

- Num. of features
- Size
- Structure
- Multi-class labels
- Missing values

Clearly, a great deal depends on the nature of the dataset, but the requirements of the analysis are also significant and may help in choosing a specific algorithm when several algorithms are tied, based on the dataset criteria. Table 1 encapsulates the criteria for the KNN, NB, DT, RF, ANN and SVM algorithms and summarizes additional advantages/disadvantages

D. HEURISTIC GUIDELINES

A generalized set of guidelines on how to proceed with categorical data are listed below. These guidelines encapsulate the previous guidance on preprocessing, encoding and algorithm selection.

Heuristic guidelines:

1. Determine what the research question/problem statement is.
2. Preprocess the dataset: Look at categories. Consolidate classes to reduce cardinality (IV.A Preprocessing Steps).
3. Select algorithm, based on the research question, problem parameters and accuracy (IV.C Table 1):

   3.1 Start simple with Naïve Bayes or KNN – does independence hold? – store the results for future comparison.

   3.1.1 If parametric requirement use NB, otherwise use KNN.

   3.2 If research question/problem statement is difficult to frame use Decision Tree and then Random Forest.

   3.3 If NB, KNN and RF results are poor AND problem is highly non-linear use SVM or ANN.

   3.3.1 If large training dataset is available use ANN.

TABLE 1. Classification algorithm criteria.

	Easily interpretable results?	**Simple Algorithm?**	**Prediction accuracy?**	**Training speed?**	**Prediction speed?**	**Parameter Tuning?**	**Small Num. of observations**	**Separates signal from noise?**	**Parametric?**	**Other adv.**	**Other disadv.**
KNN	Yes	Yes	Low	Fast	Depends on sample size	Minimal	No	No	No	No training period. Easy to implement.	Not suited to high dimensionality. Unsuitable for noisy data.
Naïve Bayes	Moderate	Moderate	Low	Fast	Fast	Moderate	Yes	Yes	Yes	Performs surprisingly well even for small datasets.	Independence may not hold.
Decision Tree	Yes	Moderate	Low	Fast	Fast	Moderate	No	No	No	Tolerates missing data. Intuitive and easy to explain.	Small changes in data lead to large tree structure changes. Can lead to overfitting of data.
Random Forest	Yes	No	High	Slow	Moderate	Moderate	No	Yes	No	Built-in feature importance facility. Tolerates missing data.	Need a large number of trees. Memory intensive.
SVM	No	No	High	Slow	Fast	Moderate	No	Yes	No	Suited to high dimensionality – even when num. dimensions exceed num. samples. Effective when there is a clear margin of separation.	Susceptible to noise. Memory intensive. Unsuitable for large datasets.
ANN	No	No	High	Slow	Fast	Moderate	No	Yes	No	Ability to train a machine. Suited to parallelization.	Not well suited to multiclass problems. Requires a large training dataset.

3.3.2 If high dimensionality use SVM.

4. Select feature encoding (IV.B Fig. 1 flowchart).
5. Train and test classifier.
6. Assess results and accuracy.

Repeat

7. If accuracy AND/OR results are poor change feature encoding AND/OR algorithm (Fig. 1 flowchart/ Table 1).
8. Examine features and eliminate outlier features.
9. Assess results and accuracy.

Until sufficient results and accuracy.

The ‘research question/problem statement’ is a general term, which could refer to establishing correlation, classification, or causation, depending on the nature of the problem. Examples could be ‘Identification of spam’, ‘making product recommendations’, ‘customer segmentation’. For this research, it was ‘the classification of domestic fire injuries’.

V. RESULTS AND DISCUSSION

A case study was used to evaluate the heuristics developed in section IV. The case study was based on a dataset of domestic fires and their resulting injuries, which present a huge challenge for emergency services (fire, medical and police). In large urban areas data is collected relating to such cases and analysis of the data will reveal patterns that can be used to prevent further cases and assist in emergency service response. Ten years of domestic fire data were examined in the case study, based on data collected for urban regions of the UK. As this dataset is categorical the heuristics developed in section IV were applied systematically to reveal useful findings to help forecast future cases and provide fire prevention strategies to the community.

A. PREPROCESSING

The data was cleaned for consistency, which involved ensuring certain terms were separated by a hyphen (a number of entries consisted of terms separated by spaces). Frequency counts were produced to show the spread of cases across the different features. Several less significant features were removed to reveal the dominant features in the dataset as:

- Gender
- Age
- Property type
- Circumstances
- The target label was 'Injury'.

The counts were also used to reduce the number of categories for several features and the 'Injury' target variable. Injuries were consolidated from 27 to 9. The Property type feature was consolidated from 19 to 7. The Circumstances feature was consolidated from 38 to 10. Age was grouped into buckets of 4 years. The research question was framed as: *How do the various features (Gender, Age, Dwelling type, Circumstances of Fire, etc.) classify the resulting injury?*

B. ALGORITHM AND ENCODING

Following the heuristic guidelines, the analysis began with the Naïve Bayes algorithm, due to its simplicity. The bnlearn python library was used, but the training accuracy was low and the algorithm did not provide a good classifier. Based on the encoding flowchart (Fig. 1), the following encodings were used for Naïve Bayes:

- Gender: Nominal encoding
- Age: Contrast encoding
- Property type: One-hot encoding
- Circumstances: One-hot encoding
- Injury: One-hot encoding

Alternative encodings (first Binary and then Target) were substituted for one-hot encoding but there was little improvement in training accuracy, suggesting the problem lay within the algorithm itself. Naïve Bayes assumes independence of features, which was unrealistic in this case. However, the results generated from bnlearn did provide some useful insights to the effects of certain features. In particular, results are shown for the different injuries according to age in Fig. 2. The graphs show injuries for adult age groups and the red bar represents females and the blue males.

The main conclusions from the age/injury plots are:

1. Younger age groups (20-35) tend to suffer from 'burns slight' due to trying to tackle the fire.
2. There are less 'burns slight' for older age groups (40-59) who tend to suffer from the 'smoke+fumes' injuries, due to being slower to evacuate premises.

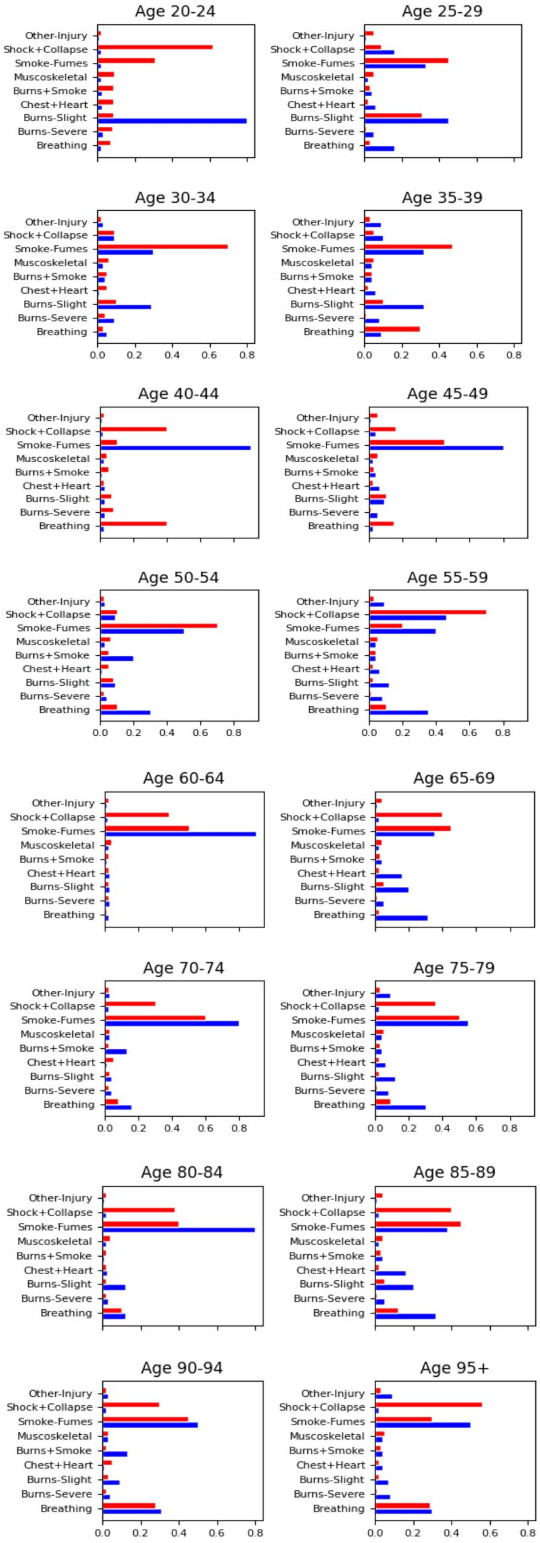


FIGURE 2. Effects of age on injury type for Naïve Bayes classifier.

3. Older age groups also suffer from 'breathing' injuries, due to other age-related health issues.
4. Females tend to suffer from 'smoke+fumes' and also 'shock-collapse' and less in terms of fighting fire injuries.

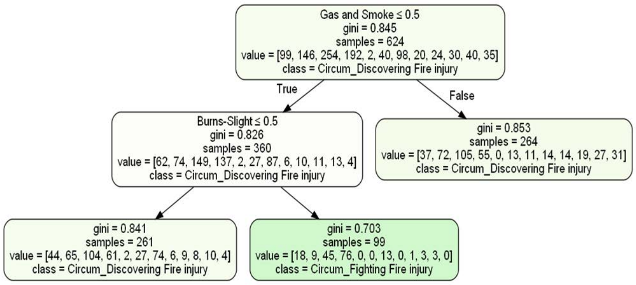

Gas and Smoke ≤ 0.5
gini = 0.845
samples = 624
value = [99, 146, 254, 192, 2, 40, 98, 20, 24, 30, 40, 35]
class = Circum_Discovering Fire injury
True
False
Burns-Slight ≤ 0.5
gini = 0.826
samples = 360
value = [62, 74, 149, 137, 2, 27, 87, 6, 10, 11, 13, 4]
class = Circum_Discovering Fire injury
gini = 0.853
samples = 264
value = [37, 72, 105, 55, 0, 13, 11, 14, 14, 19, 27, 31]
class = Circum_Discovering Fire injury
gini = 0.841
samples = 261
value = [44, 65, 104, 61, 2, 27, 74, 6, 9, 8, 10, 4]
class = Circum_Discovering Fire injury
gini = 0.703
samples = 99
value = [18, 9, 45, 76, 0, 0, 13, 0, 1, 3, 3, 0]
class = Circum_Fighting Fire injury

FIGURE 3. **Decision tree excerpt for random forest classifier for domestic fire injuries.**

The analysis then proceeded, according to the heuristics, and Random Forest was selected as the second algorithm. This required changes to the feature encodings. The initial feature encodings were selected as below, although several alternate encodings were investigated to optimize training accuracy and minimize loss (as discussed in section V.C).

- Gender: Nominal encoding
- Age: Contrast encoding
- Property type: Target encoding
- Circumstances: Target encoding
- Injury: Target encoding

The scikit-learn library was used to implement the Random Forest classifier. The algorithm was implemented with n=100 trees up to n=1000 trees and was seen to converge around n=900. Decision trees from the forest were large and a typical excerpt from a single decision tree is shown in Fig. 3.

The nodes in the tree represent the different injuries for different circumstances, such as a discovering fire injury ('Circum_Discovering Fire injury'). Each node contains useful information. First is the Gini impurity value, second is the number of samples associated with the node and third is a vector (values) which represent the number of samples in each category.

The RF algorithm provides greater insight to the research question with the added bonus of the 'feature importance' breakdown (Fig. 4).

These feature importance scores are useful and can be used in a range of situations in a predictive modeling problem to help better understand the data and the classifier. The feature importance breakdown can also help in reducing the number of input features. The feature scores highlight the most important and least important features in relation to the target label (Injury in this case). They provide invaluable information that can be interpreted by a domain expert and can also serve as the basis for gathering more data or different data for future datasets.

Several useful conclusions can also be drawn from Fig. 4:

1. The majority of injuries stem from young/middle-aged males (25-55) fighting fires.
2. Injuries are dominant in 2-3 storey single occupancy premises.
3. Circumstances involving the discovery of a fire or entrapment due to smoke are also significant.

This information proves useful for fire and other emergency services, tasked with dealing with domestic fire injuries and also fire prevention strategies. In particular,

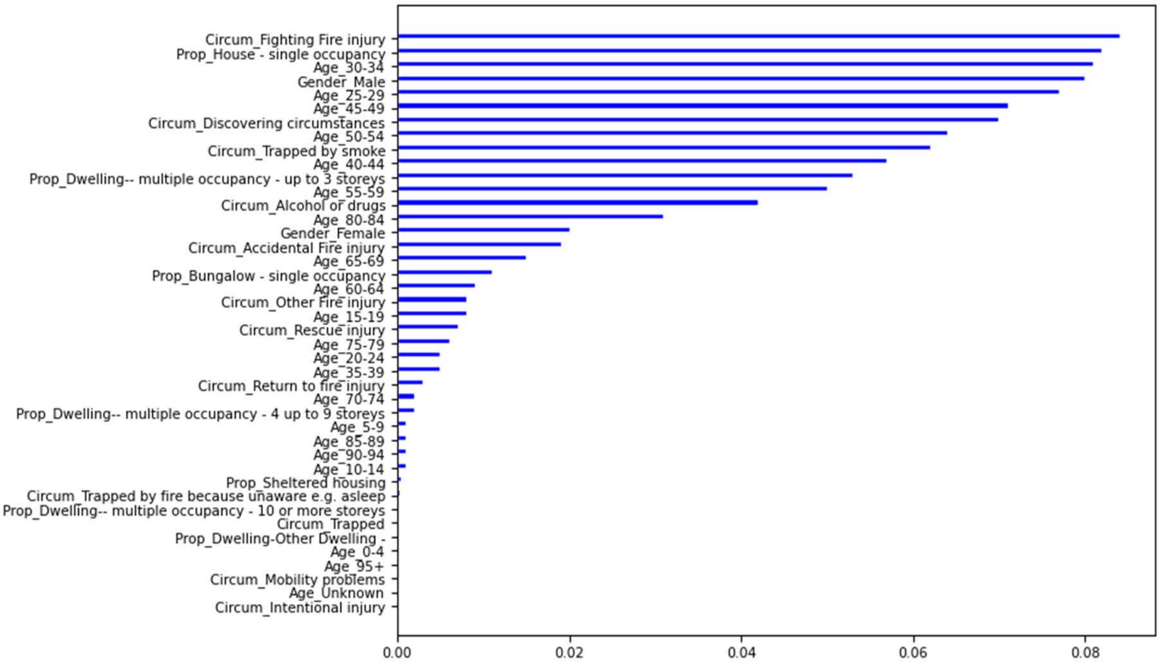


FIGURE 4. **Decision tree excerpt for random forest classifier for domestic fire injuries.**

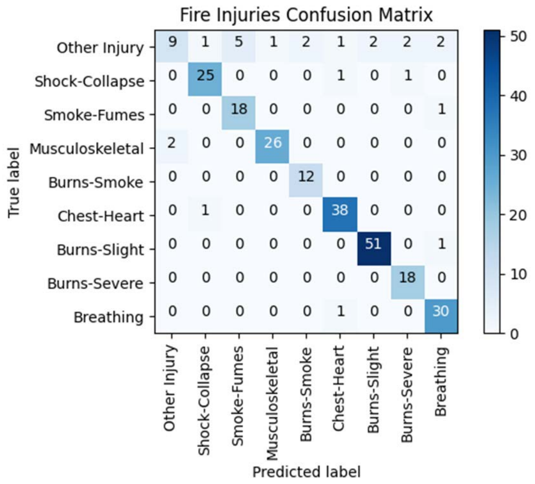


FIGURE 5. Confusion matrix for random forest classifier of domestic fire injuries.

it allows fire prevention strategists to visit at-risk groups to ensure that smoke alarms are fitted and working and escape exits are available in the event of a fire.

The results of the analysis highlight specific types of property and specific age groups for which domestic fires and certain types of fire injury are more common. The findings of the work will be used to concentrate resources at local levels, such as house-to-house visits for at-risk age groups and home checks for at-risk property types to ensure there are no fire hazards within the home (e.g. electrical wiring, open fires).

In particular, the analysis highlighted that low-rise properties are at a greater risk than high-rise properties. This reflects the fire safety assessment of different property types. High-rise buildings are required to provide a certificate of fire-safety assessment, which is not required for low-rise property types. An action resulting from the analysis would be to ensure that all new-build low-rise properties undergo some form of fire safety assessment.

C. PERFORMANCE

The performance of the RF classifier is summarized by the confusion matrix in Fig. 5. Correct prediction is 87% with the majority of false positives occurring for the 'Other Injury' category.

The RF model was also used to highlight the effects of encoding on training accuracy, convergence and loss. The RF model converged at around 900 trees and Fig. 6 shows the training accuracy and loss for different encoding techniques.

As highlighted in Fig. 6, one-hot encoding leads to significantly reduced accuracy for model training and far greater loss, as expected for decision tree algorithms. Target encoding, James-Stein encoding and Feature Hashing encoding give comparable results in terms of accuracy, with Target encoding giving slightly better results in terms of training accuracy and loss.

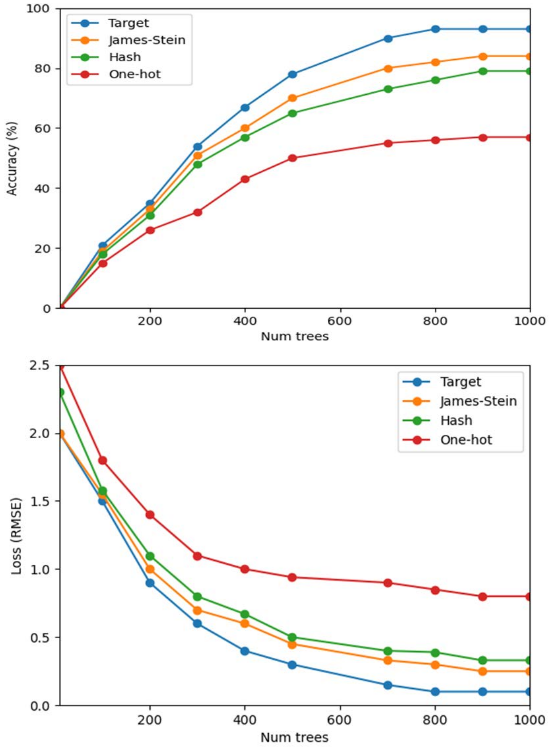


FIGURE 6. Random forest classifier training accuracy and loss for different feature encodings.

VI. CONCLUSION

This research has considered the application of machine learning techniques to datasets with a substantial categorical feature base. This area is often overlooked as machine learning was originally intended for numerical datasets. Inevitably, categorical data exists in the datasets of today, moreso in some sectors of society than others (e.g. medicine and social media).

It is crucial that systematic techniques and heuristics are available to deal with categorical datasets in order to extract the wealth of information that they contain. In sectors such as medicine, categorical datasets may hide (or mask) crucial information. If appropriate techniques are not used, it may not be possible to see trends and make predictions or forecasts in relation to patient treatments or wellbeing.

The work has made several contributions to knowledge: First the analysis of different encoding techniques, resulting in a flowchart. Second, is the analysis of the different classification algorithms in terms of their suitability for categorical data classification. Third, the heuristic guidelines, combine the former to allow categorical data analysis to take place systematically. Fourth, a realistic case study to demonstrate the application of the methods discussed.

The case study highlighted how it may be necessary to develop more than one classifier to extract meaningful data and also how different classifiers may offer different perspectives. The case study also highlighted the importance of the choice of encoding technique. In particular, how different feature encodings can affect the training accuracy, especially if a decision tree algorithm is used. In terms of algorithms, Naïve Bayes can prove useful if independence amongst features holds, whereas RF provides a fast accurate model for moderately sized datasets and a useful feature importance facility.

The case study has also highlighted how the results can be fed back to inform the collection of future datasets. For example, 'Injury' and 'Property type' we recorded as nominal variables, but the enumeration of these variables, based on an ordinal scheme may reveal further findings and improvements in classification accuracy.

Finally, the findings from the case study will be of use in assisting emergency services and their associated support personnel when dealing with domestic fire incidents. Several significant results emerged in relation to age, gender, property type and the circumstances associated with domestic fire injuries. These results will help in terms of incident planning and fire prevention strategies.

REFERENCES

[1] A. Agresti, *Categorical Data Analysis*. Hoboken, NJ, USA: Wiley, 2003.

[2] N. Bogduk and M. Stojanovic, "Group data or categorical data for outcomes of pain treatment?" *Pain Med.*, vol. 21, no. 10, pp. 2046–2052, Oct. 2020.

[3] S. Renjith, A. Sreekumar, and M. Jathavedan, "An empirical research and comparative analysis of clustering performance for processing categorical and numerical data extracts from social media," *Acta Scientiarum. Technol.*, vol. 44, no. 1, Mar. 2022, Art. no. e58653, doi: 10.4025/actascitechnol.v44i1.58653.

[4] B. Ru, A. Alvi, V. Nguyen, M. A. Osborne, and S. Roberts, "Bayesian optimisation over multiple continuous and categorical inputs," in *Proc. 37th Int. Conf. Mach. Learn., Mach. Learn. Res.*, vol. 119, 2020, pp. 8276–8285. [Online]. Available: https://proceedings.mlr.press/v119/ru20a.html

[5] P. Cerda and G. Varoquaux, "Encoding high-cardinality string categorical variables," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 3, pp. 1164–1176, Mar. 2022, doi: 10.1109/TKDE.2020.2992529.

[6] M. K. Dahouda and I. Joe, "A deep-learned embedding technique for categorical features encoding," *IEEE Access*, vol. 9, pp. 114381–114391, 2021, doi: 10.1109/ACCESS.2021.3104357.

[7] J. Ji, W. Pang, Z. Li, F. He, G. Feng, and X. Zhao, "Clustering mixed numeric and categorical data with cuckoo search," *IEEE Access*, vol. 8, pp. 30988–31003, 2020, doi: 10.1109/ACCESS.2020.2973216.

[8] Z. Ruiz-Chavez, J. Salvador-Meneses, and J. Garcia-Rodriguez, "Machine learning methods based preprocessing to improve categorical data classification," in *Intelligent Data Engineering and Automated Learning* (Lecture Notes in Computer Science), vol. 11314. Cham, Switzerland: Springer, 2018, pp. 297–304, doi: 10.1007/978-3-030-03493-1_32.

[9] S. Hamad, "Creativity: Method or magic?" in *Consciousness and Cognition*. New York, NY, USA: Academic, Jan. 2007, pp. 127–137.

[10] H. Tahami and H. Fakhravar, "A literature review on combining heuristics and exact algorithms in combinatorial optimization," *Eur. J. Inf. Technol. Comput. Sci.*, vol. 2, no. 2, pp. 6–12, 2022, doi: 10.24018/compute.2022.2.2.50.

[11] R. Marte, *Handbook of Heuristics*. Cham, Switzerland: Springer, 2018.

[12] Z. Lu and M. Yin, "Human reliance on machine learning models when performance feedback is limited: Heuristics and risks," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, May 2021, pp. 1–16.

[13] P. Cerda, G. Varoquaux, and B. Kégl, "Similarity encoding for learning with dirty categorical variables," *Mach. Learn.*, vol. 107, pp. 1477–1494, Sep. 2018, doi: 10.1007/s10994-018-5724-2.

[14] J. T. Hancock and T. M. Khoshgoftaar, "Survey on categorical data for neural networks," *J. Big Data*, vol. 7, no. 1, pp. 1–41, Dec. 2020, doi: 10.1186/s40537-020-00305-w.

[15] K. Potdar, T. S. Pardawala, and C. D. Pai, "A comparative study of categorical variable encoding techniques for neural network classifiers," *Int. J. Comput. Appl.*, vol. 175, no. 4, pp. 7–9, 2017.

[16] D. Chicco, "Ten quick tips for machine learning in computational biology," *BioData Mining*, vol. 10, no. 1, pp. 1–17, Dec. 2017, doi: 10.1186/s13040-017-0155-3.

[17] R.-C. Chen, C. Dewi, S.-W. Huang, and R. E. Caraka, "Selecting critical features for data classification based on machine learning methods," *J. Big Data*, vol. 7, no. 52, pp. 1–26, Dec. 2020, doi: 10.1186/s40537-020-00327-4.

[18] Q. Zhang, J. Xu, L. Xu, and H. Guo, "Deep convolutional neural networks for forest fire detection," in *Proceedings of the International Forum on Management, Education and Information Technology Application* (Advances in Social Science, Education and Humanities Research). Amsterdam, The Netherlands: Atlantis Press, 2016, pp. 568–575.

[19] X. Yan, H. Cheng, Y. Zhao, W. Yu, H. Huang, and X. Zheng, "Real-time identification of smoldering and flaming combustion phases in forest using a wireless sensor network-based multisensor system and artificial neural network," *Sensors*, vol. 16, no. 8, pp. 1–10, 2016.

[20] M. Saoudi, A. Bounceur, R. Euler, and T. Kechadi, "Data mining techniques applied to wireless sensor networks for early forest fire detection," in *Proc. Int. Conf. Internet Things Cloud Comput.*, Mar. 2016, pp. 1–7.

[21] A. Karouni, B. Daya, and P. Chauvet, "Applying decision tree algorithm and neural networks to predict forest fires in Lebanon," *J. Theor. Appl. Inf. Technol.*, vol. 63, no. 2, pp. 282–291, 2014.

[22] Z. S. Pourtaghi, H. R. Pourghasemi, R. Aretano, and T. Semeraro, "Investigation of general indicators influencing on forest fire and its susceptibility modeling using different data mining techniques," *Ecol. Indicators*, vol. 64, pp. 72–84, May 2016.

[23] A. Alkhatib, "A review on forest fire detection techniques," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 3, pp. 1673–1683, 2014.

[24] M. Castelli, L. Vanneschi, and A. Popovič, "Predicting burned areas of forest fires: An artificial intelligence approach," *Fire Ecol.*, vol. 11, no. 1, pp. 106–118, Apr. 2015.

[25] X. Zhao, R. Lovreglio, and D. Nilsson, "Modelling and interpreting pre-evacuation decision-making using machine learning," *Autom. Construct.*, vol. 113, May 2020, Art. no. 103140.

[26] K. Wang, K. Shi, A. P. Goh, and S. Qian, "A machine learning based study on pedestrian movement dynamics under emergency evacuation," *Fire Saf. J.*, vol. 1, no. 106, pp. 76–163, 2019.

[27] I. Gomaa, M. Adelzadeh, S. Gwynne, B. Spencer, Y. Ko, N. Bénichou, C. Ma, N. Elsagan, D. Duong, E. Zalok, and M. Kinateder, "A framework for intelligent fire detection and evacuation system," *Fire Technol.*, vol. 57, no. 6, pp. 3179–3185, 2021.

[28] S. Sarkar, S. Vinay, R. Raj, J. Maiti, and P. Mitra, "Application of optimized machine learning techniques for prediction of occupational accidents," *Comput. Oper. Res.*, vol. 106, pp. 210–224, Jun. 2019.

[29] M. Madaio, O. L. Haimson, W. Zhang, X. Cheng, M. Hinds-Aldrich, B. Dilkina, and D. H. P. Chau, "Identifying and prioritizing fire inspections: A case study of predicting fire risk in Atlanta," in *Proc. Bloomberg Data Good Exchange Conf.*, New York, NY, USA: Bloomberg, 2015.

[30] C. K. Lau, K. K. Lai, Y. P. Lee, and J. Du, "Fire risk assessment with scoring system, using the support vector machine approach," *Fire Saf. J.*, vol. 78, pp. 188–195, Nov. 2015.

[31] H. Bengtsson, "Development of a framework for application of Bayesian networks in fire safety engineering in Denmark," M.S. thesis, Univ. Stavanger, Stavanger, Norway, 2014.

[32] L. Surya, "Risk analysis model that uses machine learning to predict the likelihood of a fire occurring at a given property," *Int. J. Creative Res. Thoughts*, vol. 5, no. 1, pp. 2320–2882, 2017.

[33] D. B. Matellini, A. D. Wall, I. D. Jenkinson, J. Wang, and R. Pritchard, "A three-part Bayesian network for modeling dwelling fires and their impact upon people and property," *Risk Anal.*, vol. 38, no. 10, pp. 2087–2104, Oct. 2018.

[34] Z. R. Maruf and A. D. Laksito, "The comparison of distance measurement for optimizing KNN collaborative filtering recommender system," in *Proc. 3rd Int. Conf. Inf. Commun. Technol. (ICOIACT)*, Nov. 2020, pp. 89–93.

[35] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[36] F. Fabris, A. Doherty, D. Palmer, J. P. de Magalhães, and A. A. Freitas, "A new approach for interpreting random forest models and its application to the biology of ageing," *Bioinformatics*, vol. 34, no. 14, pp. 2449–2456, Jul. 2018, doi: 10.1093/bioinformatics/bty087.

[37] M. Kuhn and K. Johnson, *Feature Engineering and Selection: A Practical Approach for Predictive Models*. Boca Raton, FL, USA: CRC Press, 2019.

[38] B. Roy. (Jul. 2019). *All About Categorical Variable Encoding. Towards Data Science*. Accessed: Mar. 24, 2022. [Online]. Available: https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02

[39] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 161–168.

[40] T. Pranckevičius and V. Marcinkevičius, "Comparison of Naive Bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification," *Baltic J. Mod. Comput.*, vol. 5, no. 2, pp. 221-232, 2017.

[41] F. Y. Osisanwo, J. E. Akinsola, O. Awodele, J. O. Hinmikaiye, O. Olakanmi, and J. Akinjobi, "Supervised machine learning algorithms: Classification and comparison," *Int. J. Comput. Trends Technol.*, vol. 48, no. 3, pp. 38–128, 2017.

DENIS REILLY received the B.Eng. degree (Hons.) in electrical and electronic engineering and the M.Sc. degree (Hons.) in computer science and software engineering from the University of Liverpool, in 1989 and 1991, respectively, and the Ph.D. degree in distributed system and middleware from Liverpool John Moores University, in 2003.

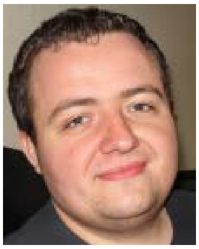
He is currently a Principal Lecturer with the School of Computer Science and Mathematics, Liverpool John Moores University. In addition to his research interests, he is a Course Leader for B.Sc. (Hons.) computer studies and B.Sc. (Hons.) computer networks. His research interests include machine learning, data analytics, distributed systems and middleware, medical informatics, and cloud computing and security.

MARK TAYLOR received the B.Sc. degree (Hons.) in mathematics from Warwick University, U.K., in 1984, and the Ph.D. degree in computer science from Salford University, U.K., in 1999.

He is currently a Senior Lecturer with the Department of Computer Science, Liverpool John Moores University, an Examiner for the British Computer Society, and an Academic Reference Group Member for Her Majesty's Inspectorate of Constabularies and Fire and Rescue Services, U.K. He has a wide and varied research background covering data analytics, statistical modeling, risk analysis, and fire prevention.

Dr. Taylor is a Chartered IT Professional, a Charted Engineer, and a Chartered Scientist.

PAUL FERGUS is currently a Professor of machine learning and the Head of the Data Science Research Centre at the School of Computer Science and Mathematics at Liverpool John Moores University. His research interests include machine learning for detecting and predicting preterm births. He is also interested in the detection of foetal hypoxia, electroencephalogram seizure classification, and bioinformatics. He is also conducting research with Mersey Care NHS Foundation Trust looking at the use of smart meters to detect activities of daily living in people living alone with dementia by monitoring the use of home appliances to model habitual behaviors for early intervention practices and safe independent living at home. He has competitively won external grants to support his research from HEFCE, Royal Academy of Engineering, Innovate U.K., Knowledge Transfer Partnership, North West Regional Innovation Fund, and Bupa. He has published over 200 peer-reviewed articles in these areas.



CARL CHALMERS is currently a Senior Lecturer with the Department of Computer Science, Liverpool John Moores University. His main research interests include the advanced metering infrastructure, smart technologies, ambient assistive living, machine learning, high-performance computing, cloud computing, and data visualization. His current research area focuses on remote patient monitoring and ICT-based healthcare. He is also leading a three-year project on smart energy data and dementia in collaboration with Mersey Care NHS Trust to monitor and model the behavior of dementia patients and facilitate safe independent living. In addition, he is also working in the area of high-performance computing and cloud computing to support and improve existing machine learning approaches, while facilitating application integration.

STEVEN THOMPSON received the B.Sc. degree (Hons.) in information systems and the M.Sc. degree (Hons.) in wireless and mobile computing from Liverpool John Moores University, in 2008 and 2013, respectively, where he is currently pursuing the Ph.D. degree.

He is currently a Senior Technical Officer with the Department of Computer Science, Liverpool John Moores University. He is also performing part-time research. His research interests include data analytics, machine learning, deep learning, and neural networks.