An Enhanced Similarity Measure–Driven K-Nearest Neighbor Framework for Categorical Data Classification

Basharat Ahmad Hassan[1], Sana Shafiq[1], Muhammad Abbas[2], Muhammad Zikriya[1], Feroz Khan[1], Zainab Ihsan[1], Tasawar Shah[1], Jamaluddin[1]
[1]Qurtaba University of Science and Information Technology, Peshawar, Pakistan
[2]Abasyn University, Peshawar, Pakistan
*__Correspondence__: basharat94@gmail.com

Citation | Hassan. B. A, Shafiq. S, Abbas. M, Zikriya. M, Khan. F, Ihsan. Z, Shah. T. Jamaluddin, "An Enhanced Similarity Measure–Driven K-Nearest Neighbor Framework for Categorical Data Classification", IJIST, Vol. 7, Issue. 4 pp 2757-2772, November 2025
Received | October 03, 2025 **Revised |** November 04, 2025 **Accepted |** November 11, 2025 **Published |** November 18, 2025.

Machine learning provides effective answers to real-world classification issues by combining supervised approaches (e.g., regression, SVMs, decision trees, neural networks) and unsupervised techniques (e.g., clustering, PCA). Comparing categorical data to numerical data reveals that the former is still understudied. This study compares three variations of the K-Nearest Neighbors (KNN) algorithm, Dice Coefficient KNN (DKNN), Overlap Coefficient KNN (OKNN), and Simple Match Coefficient KNN (SMKNN) on three categorical datasets: Malware Detection, Hospital Readmission (Kaggle) and Mushroom (UCI Repository). Each variation improves classification performance by incorporating a unique similarity metric. Recall, accuracy, precision, and F1-score were used to evaluate the models. According to experimental results, SMKNN consistently performed better than the other variations, obtaining an average F1-score of 93.3%, accuracy of 88.29%, precision of 89.33%, and recall of 98%. With an F1-score of 91% and an average accuracy of 83.89%, OKNN came in second, while DKNN did worse with an accuracy of 73.74%. These results demonstrate the stability and promise of SMKNN as a dependable model for categorical data classification, highlighting its exceptional and flexible performance across a variety of datasets. The study gives useful information for identifying the best KNN variations for data-driven applications.

Keywords: Machine Learning, Classification, KNN, DKNN, OKNN, SMKNN.

Introduction:

Categorical data refers to data composed of distinct categories or labels that represent specific/ limited values that can be assumed. Categorical data comprises non-numeric values that represent distinct groups without any inherent order or quantitative relationship. Examples include variables like gender, color, or type of product etc. In categorical data, categories can be nominal, meaning they have no inherent order, or ordinal, meaning they follow a specific order or ranking. The primary objective in analyzing categorical data is to examine how categories are distributed and to explore the relationships between different categorical variables [1]. **Categorical data** is generally classified into two main types: nominal and ordinal. **Nominal data:** Nominal data are a type of categorical data that have no inherent order or ranking [2] e.g., hair color, eye color, and types of food. **Ordinal data:** Ordinal data represents a category of categorical data characterized by a built-in order or ranking e.g., educational levels (high school, bachelor's, master's, Doctor of Philosophy) and customer satisfaction ratings (unsatisfied, neutral, satisfied, highly satisfied). In addition, there is a special subtype known as binary data, which is a form of nominal data consisting of only two possible categories e.g., variables such as yes/no, male/female, and true/false. Classification of categorical data in the machine learning field is always challenging; there are many classification techniques used for categorical data.

Classification is a type of supervised machine learning task in which the goal is to predict the categorical class or label of a given input based on learned patterns from training data [3]. The algorithm is trained on a labelled dataset that includes inputs and their respective class labels. The training process involves finding the relationship between the inputs and the class labels, which is then used to make predictions on new data. Classification algorithms are applied across a wide range of domains, including spam detection, sentiment analysis, image classification, and disease diagnosis. Their performance is commonly evaluated using metrics such as accuracy, precision, recall, and F1-score. There are numerous classification techniques in machine learning, each with its own strengths and limitations.

K-Nearest Neighbors (KNN) is highly sensitive to irrelevant or noisy features, as it treats all features equally when computing distances. This can negatively impact performance, especially if irrelevant features disproportionately influence the distance calculations. **Importance of Feature Scaling**: KNN depends on measuring distances between data points, so it is essential to scale features. Without scaling, features with larger numerical ranges can disproportionately influence the results, leading to biased predictions. KNN is based on distances, and features with larger scales can dominate the distances. **Curse of dimensionality**: KNN can suffer from the curse of dimensionality, which means that the performance decreases when the number of features increases. This is because the distances between data points become less meaningful in high-dimensional spaces, making it harder to find the nearest neighbors. **Hyper-parameter tuning**: The number of neighbors (K) is very important and needs to be specified before training the model. Choosing the exact value for K is vital for the good performance of the model, and finding the optimal value can be difficult, especially in complicated datasets [4]. KNN is not inherently suitable for handling categorical features, as it relies on distance calculations to make predictions. While categorical features can be converted into numerical values through encoding, this approach may result in a loss of information and increase the complexity of the algorithm.

Similarity measure is a method used in ML to quantify the similarity or dissimilarity between two objects, such as data points or features [5]. The similarity measure is employed to identify the nearest or most similar objects to a target object, utilizing a pre-defined metric of similarity. Several frequently employed similarity measures in machine learning are: **Euclidean Distance**: E-D is a direct line distance between two points in a multi-dimensional space. It is a simple and widely used similarity measure in K-Nearest Neighbors (KNN)

algorithms [6]. **Manhattan Distance**: M-D can also called the taxicab distance, is the sum of the absolute differences among the coordinates of given points in a multi-dimensional space. It is not as sensitive to outliers as the Euclidean distance [7]. **Cosine Similarity** (C-S): Cosine similarity measures the similarity between two non-zero vectors in an inner product space by calculating the cosine of the angle between them. Its values range from 1 to -1, where 1 indicates perfect similarity, 0 indicates no similarity, and -1 represents complete dissimilarity [8].

The conventional application of the KNN classifier, primarily utilized for numerical or continuous data, which is extended to cater specifically to categorical data. Unlike its traditional usage with distance-based similarity metrics such as Manhattan and Euclidean distances, the adaptation involves the integration of a novel approach. Categorical-based similarity measures, including the Simple Matching coefficient, Dice coefficient, and Overlap coefficient, are employed. This integration allows the KNN classifier to effectively navigate and analyze datasets characterized by categorical variables, providing an efficient framework for classification tasks in contexts where traditional numerical approaches may not be suitable. The utilization of these categorical similarity measures enhances the model's ability to discern patterns and relationships within non-numeric data, thereby broadening its applicability across diverse domains.

Literature Review:

Categorical data categorization remains one of the most difficult areas in machine learning, as traditional methods are largely built for numerical data. Numeric distance metrics, such as the Euclidean or Manhattan distance, are insufficient for categorical qualities, which lack inherent ordering and continuity. To address these difficulties, academics have proposed a variety of clustering methods, similarity metrics, and classification frameworks with the goal of improving categorical data accuracy and interpretability.

In reference [9], it made an early contribution by creating the k-modes algorithm, which extended k-means to include categorical data by employing a frequency-based dissimilarity metric. Although this strategy proved scalable and effective, it was sensitive to startup and did not explore alternative distance measurements. [10] expanded on this idea by presenting a k-means algorithm capable of handling both numerical and categorical input. Their study demonstrated linear complexity and interpretability via attribute contribution; however, it was only evaluated on a few datasets. [11] introduced projected clustering for high-dimensional categorical data and proposed novel validation indices, although their research lacked information on initialization and optimal cluster determination. Similarly, [12] introduced the Clicks algorithm, stressing scalability and increased performance; however, the dataset diversity and parameter tweaking were not properly documented. These early clustering approaches laid the groundwork for categorical data management, but they were hampered by tight similarity criteria and limited generalization.

Researchers then concentrated on improving similarity measurements to improve clustering and classification results. [13] introduced a Generalized Similarity Metric (GSM) that is combined with the ROCK method, providing a unified approach to categorical similarity calculation. Despite their contributions, parameterization and comparative analysis were restricted. [14] created Maximal Resemblance Data Labeling, a novel representative measure for categorical clusters, but empirical validation was insufficient. [15] developed a multi-viewpoint similarity measure that increased clustering quality but did not provide computational specifics. [16] proposed a semantic-based similarity measure that employs ontology and domain taxonomy to capture deeper links between attributes; nevertheless, the method was computationally complex. [17] introduced HeteSim, a relevance measure for heterogeneous entities, which performed well in non-machine-learning tasks but had limited comparative evaluation. Collectively, these studies show that, while enhanced similarity

measures improve interpretability and local accuracy, many are yet unproven across domains and big datasets.

Other machine-learning techniques have been employed to handle categorical data. [18] examined Random Forest, Naïve Bayes, and KNN for health classification, concluding that ensemble techniques increase prediction but require better data processing. [19] demonstrated the extensive applicability of KNN, Naïve Bayes, and Decision Trees, but did not cover algorithmic optimization in detail. [20] investigated categorical variable encoding in neural networks and found improved classification, although with dataset-specific restrictions. [21] offered a local anomaly detector that obtained high precision but required sophisticated parameter adjustment, whereas [22] proposed the Learning-Based Dissimilarity technique, which demonstrated robustness to noise and high dimensionality despite being tested on a small number of datasets. These works demonstrate a trend toward hybrid and learning-based frameworks, yet they frequently suffer from generalization and computational issues.

Deep learning has been used to capture complex correlations between category characteristics. [23] used convolutional and recurrent neural networks for intrusion detection, outperforming standard models but with limited experimental transparency. Similarly, [23] employed a fully convolutional network for segmentation, which achieved high simulation accuracy but lacked generalization. These investigations demonstrate that deep models can improve performance, but they are computationally intensive and difficult to comprehend, limiting their practical application in categorical applications.

Recent research on distance and similarity measurements confirms their essential relevance. [24] employed Hamming Distance to detect malware, demonstrating its reliability for binary-categorical data but without feature rationale. [25] developed the TaxMap clustering mechanism, which combines many similarity measures, although the internal processes were not adequately explained. [5] used the Dice Coefficient and bootstrapping for threshold analysis, which improved accuracy but added complexity. These results suggest a progressive shift toward tailored, domain-specific similarity measures that strike a balance between robustness and interpretability.

Despite this development, significant research gaps remain. Most KNN-based models use fixed similarity metrics that do not sufficiently capture categorical dependencies [3][26]. Many strategies are evaluated solely on standard datasets, such as UCI, which limits their real-world application [22][10]. Hybrid and deep-learning frameworks improve accuracy while increasing computing costs and reducing interpretability [27][16]. Few systems dynamically adjust similarity weights or maximize feature importance, and most studies ignore transparency in classification judgments, which is critical in sensitive domains like healthcare and education [18].

Overall, the literature examined demonstrates that existing clustering, similarity, and classification algorithms for categorical data provide useful insights but are still incomplete. Clustering methods lay the groundwork for categorical analysis, while similarity metrics improve interpretability, and deep-learning models provide representational capability. However, none address accuracy, flexibility, and interpretability simultaneously. These shortcomings indicate the need for an improved similarity-measure-driven K-Nearest Neighbor framework that can adapt distance computations to categorical data characteristics while being robust and computationally efficient.

Proposed Research Methodology:

The proposed research methodology is illustrated in a flowchart, outlining the entire research process from initiation to completion. The research process flow chart is shown below in Figure 1.

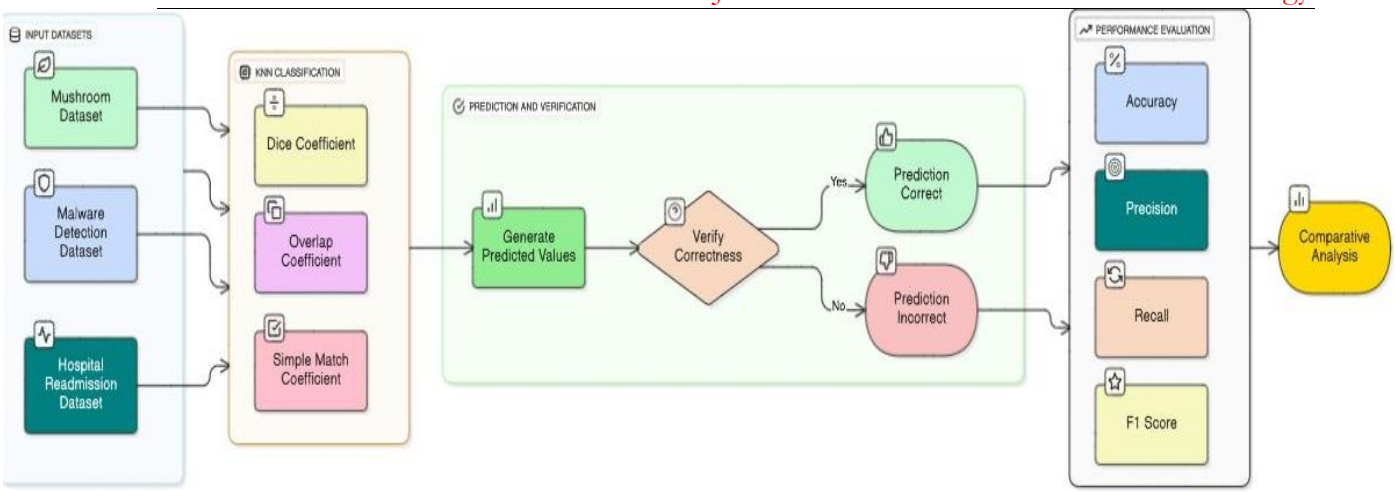

Figure 1. Research Process Flow Chart

Datasets Description:

Three varied benchmark datasets representing various domains were used to rigorously evaluate the proposed KNN-based framework for categorical data classification. Each dataset has a unique category of characteristics and levels of complexity, allowing for a thorough evaluation of the similarity measures used.

Mushroom Dataset:

This dataset contains categorical variables collected from software behavior and code structure to help distinguish between malicious and benign software samples. Its categorical character and large dimensionality make it a useful benchmark for assessing the robustness of similarity-based KNN algorithms in cybersecurity applications.

Malware Detection Dataset:

This dataset contains categorical variables collected from software behavior and code structure to help distinguish between malicious and benign software samples. Its categorical character and large dimensionality make it a useful benchmark for assessing the robustness of similarity-based KNN algorithms in cybersecurity applications.

Hospital Readmission Dataset:

This dataset includes anonymized electronic health records with categorical information like diagnosis codes, admission kinds, and treatment outcomes. The target property specifies whether a patient was readmitted within a defined time frame, posing a difficult medical classification problem due to inherent feature heterogeneity and category interdependence.

K-Nearest Neighbors (KNN) Classifier:

The k-NN algorithm is one of the simplest learning algorithms in machine learning. Its primary limitation is its sensitivity to local data structures, which can lead to decreased performance in certain situations. The KNN classifier is chosen because it has been widely applied across various datasets and has consistently demonstrated strong performance. K-nearest neighbors is a classification method that predicts the conditional distribution of Y given X and allocates observations to classes with the highest probability. Given a specified positive integer K, the algorithm identifies the K observations nearest to the test instance. x_0. It then estimates the conditional probability of x_0 belonging to class j using the corresponding formula [28].

$$P(y = j \mid x0) = (Cj / K) \quad (1)$$

where N0 is the set of K nearest observations and I (yi=j) evaluates to 1 if a particular observation (xi, yi) in N0 is a member of class j, and is an indicator variable that evaluates to

0 otherwise. Case. After estimating these probabilities, K-nearest neighbors assign observation x0 to the class with the largest prior probability.

Similarity Measures:

Dice Coefficient:

The Dice coefficient is a similarity measure commonly used in image processing and computer vision to assess the overlap between two binary sets, such as segmented images. It is also referred to as the Sørensen–Dice coefficient or Dice similarity coefficient [29].

The Dice coefficient is defined mathematically as:

$$\text{Dice}(A, B) = 2|A \cap B| / (|A| + |B|) \quad (2)$$

where A and B are two binary sets, and $|A|$ and $|B|$ represent the number of elements in sets A and B, respectively. The formula calculates the size of the intersection between the two sets relative to the total size of the combined sets. The Dice coefficient ranges from 0 to 1, with 1 indicating that the sets are alike and 0 indicating that there is no overlap between the sets.

Overlap Coefficient:

The Overlap Coefficient is a similarity measure among two sets that counts the size of the intersection in relation to the size of the union of the sets [29]. The mathematical equation for the Overlap Coefficient is given by:

$$\text{OC}(A, B) = |A \cap B| / \min(|A|, |B|) \quad (3)$$

where A and B are the two sets being compared, $|A|$ and $|B|$ are the sizes of the sets, and $|A \cap B|$ is the size of their intersection. The value of the Overlap Coefficient ranges from 0 to 1, with 1 showing that the sets are similar and 0 indicating that they have no elements the common.

Simple Match Coefficient:

Simple Match Coefficient (SMC) is a similarity measure between two binary data sets. It measures the similarity between two sets as the size of the intersection of the sets divided by the size of the union of the sets [29].

Mathematically, the SMC can be defined as:

$$\text{SMC}(A, B) = |A \cap B| / |A \cup B| \quad (4)$$

where A and B are two sets and $|A|$ represents the number of elements in set A, $\cap$ and $\cup$ represent the intersection and union of the sets, respectively.

Model Architectures:

DKNN (Dice Coefficient with KNN):

An integrated formula for using the Dice coefficient as a similarity metric in the k-Nearest Neighbors (KNN) algorithm [24].

Integrated KNN Formula with Dice Coefficient:

$$\hat{y} = MajorityVote(Neighbors_{Dice}(x_{test,}k)) \quad (5)$$

Where:

$\hat{y}$ is the predicted class label.

$\text{Neighbors}_{\text{Dice}}(x_{test,}k)$ is the set of k nearest neighbors of the test instance $\boldsymbol{x_{test,}}$ Based on the Dice coefficient as the similarity metric.

The Dice coefficient itself is used within the calculation of similarities when determining the nearest neighbors. The similarity between two instances x_i and x_j Using Dice coefficient can be defined as [23].

$$Similarity_{Dice}(x_j, x_j) = \frac{2\times|Features_{x_i} \cap Features_{x_j}|}{|Features_{x_i} + Features_{x_j}|} \quad (6)$$

In the KNN algorithm, this similarity metric is used to identify the nearest neighbors, and the majority voting scheme is then applied to determine the predicted class label [23].

OKNN (Overlap Coefficient with KNN):

The Overlap coefficient as a similarity metric in the k-Nearest Neighbors (KNN) algorithm [23].

Integrated KNN Formula with Overlap Coefficient:

$$\hat{y} = MajorityVote(Neighbors_{Overlap}(x_{test,}k)) \quad (7)$$

Where:

$\hat{y}$ is the predicted class label.

$\text{Neighbors}_{\text{Overlap}}(\text{x}_{\text{test,}}\text{k})$ is the set of k nearest neighbors of the test instance $\boldsymbol{x_{test,}}$ Based on the Overlap coefficient as the similarity metric.

The Overlap coefficient itself is used within the calculation of similarities when determining the nearest neighbors. The similarity between two instances x_i and x_j The Overlap coefficient can be defined as [23].

$$\text{Similarity}_{\text{Overlap}}(\text{x}_\text{j}, \text{x}_\text{j}) = \frac{|\,\text{Features}_{\text{x}_\text{i}} \cap \text{Features}_{\text{x}_\text{j}}\,|}{\min(|\,\text{Features}_{\text{x}_\text{i}}, \text{Features}_{\text{x}_\text{j}}\,|)} \quad (8)$$

SMKNN (Simple Match Coefficient with KNN):

Simple Matching Coefficient as a similarity metric in the k-Nearest Neighbors (KNN) algorithm [30].

Integrated KNN Formula with Simple Matching Coefficient:

$$\hat{y} = MajorityVote(Neighbors_{SimpleMatch}(x_{test,}k)) \quad (9)$$

Where:

$\hat{y}$ is the predicted class label.

$\text{Neighbors}_{\text{SimpleMatch}}(\text{x}_{\text{test,}}\text{k})$ is the set of k nearest neighbors of the test instance $\boldsymbol{x_{test,}}$ Based on the Simple Match coefficient as the similarity metric.

The Simple Match coefficient itself is used within the calculation of similarities when determining the nearest neighbors. The similarity between two instances x_i and x_j The Simple Match coefficient can be defined as [30].

$$Similarity_{SimpleMatch}(x_j, x_j) = \frac{|\,Features_{x_i} \cap Features_{x_j}\,|}{min(|\,Features_{x_i} \cup Features_{x_j}\,|)} \quad (10)$$

Pseudo code of Proposed Integrated Predictive Model:

Begin.
Step 1: The Categorical dataset is loaded.
Step 2: The Dataset is divided into training and testing subsets.
Step 3: SMKNN, OKNN, and DKNN classifiers are initialized with predefined k values.
Step 4: Classifiers are trained on the training subset.
Step 5: Empty prediction lists are created for SMKNN, OKNN, and DKNN.
Step 6: For each test instance in X_test:
Step 6.1: SMC similarity scores between test and training instances are computed.
Step 6.2: OC similarity scores between test and training instances are computed.
Step 6.3: DC similarity scores between test and training instances are computed.
Step 6.4: Top-k most similar instances are identified for each coefficient.
Step 6.5: Corresponding class labels of top-k instances are obtained.
Step 6.6: The Majority class among the top-k neighbors is selected as the prediction.
Step 6.7: Prediction is added to the respective list.
Step 7: Accuracy and related metrics are calculated for all classifiers.
Step 8: Performance comparison of SMKNN, OKNN, and DKNN is presented.
End.

In the context of this research , a comprehensive experimental setup was established to assess the performance of three distinct machine learning models: DKNN, OKNN, and SMKNN. The experiments were carried out on an HP EliteBook 840 G1 laptop, featuring a 4th-generation Intel Core i5 processor, 8GB of DDR3 RAM, a 128GB SSD for fast data access and program execution, and a 500GB HDD for additional data storage. The models were implemented and executed on three different datasets using PyCharm, a widely used Python integrated development environment (IDE). The chosen datasets are carefully selected to represent a diverse range of scenarios and challenges relevant to my research objectives. Before conducting the experiments, essential data preprocessing steps were carried out, including data cleaning, feature selection, and partitioning into training and testing sets. The models were implemented in Python within PyCharm and executed while closely monitoring resource usage and execution times. Afterward, we thoroughly analyzed the results, assessing model performance using key metrics like accuracy, recall, precision, and F1-score, and presented these findings with informative visualizations. This systematic experimental approach enables a thorough evaluation of the models, providing insights into their suitability and effectiveness within the specific context of this research.

Dataset Description:

In this research, we utilized three diverse datasets for our machine learning experiments. The first dataset, sourced from Kaggle, was focused on Malware Detection, challenging us to develop and evaluate models capable of identifying malicious software. Another dataset from Kaggle on Hospital Readmissions offered comprehensive healthcare data, enabling the exploration of predictive modeling to better understand and improve patient readmission outcomes. Finally, the Mushroom Dataset from the UCI Machine Learning Repository was used to study classification algorithms for distinguishing between edible and poisonous mushrooms, highlighting the practical importance of accurate identification for food safety.

Table 1. Dataset Specifications

Dataset Name	Number of Attributes	Number of Instances	Sourced From
Malware Detection	15	373	Kaggle
Hospital readmission	14	2001	Kaggle
Mushroom Dataset	9	3001	UCI Repository

These features were derived from a combination of binary, hexadecimal, and DLL (Dynamic Link Library) call analyses of Windows executables. The dataset comprises 373 samples in total, with 301 labeled as malicious and 72 as non-malicious. Notably, the dataset is imbalanced, containing a larger proportion of malware samples compared to benign files. The dataset includes 531 features, denoted as F_1 through F_531, alongside a label column that indicates whether a file is malicious or non-malicious. To simplify feature representation, the binary, hexadecimal, and DLL call features were labeled sequentially as F_1, F_2, and so on. This approach was adopted due to the complexity of directly representing the original feature names. Additionally, some of the 531 features may have limited significance and could potentially be dropped during feature engineering. Ultimately, the 'label' column serves as the ground truth, clearly indicating whether each executable file is classified as malware or non-malicious. Further exploration and feature engineering may uncover valuable insights from this dataset. We have taken 15 attributes and 373 instances in our research [31].

The Hospital Readmission dataset, comprising 14 attributes, including 'race,' 'gender,' 'age,' 'admission_type_id,' 'discharge_disposition_id,' 'admission_source_id,' 'time_in_hospital,' 'num_lab_procedures,' 'num_procedures,' 'num_medications,' 'number_outpatient,' 'number_emergency,' 'number_inpatient,' 'number_diagnoses,' and a class label 'readmitted,' presents a valuable resource for healthcare analytics and predictive modeling. With a sizable sample size of 59558 instances, this dataset offers an ample view of

patient interactions within healthcare systems. These attributes encompass a broad range of patient information, ranging from demographic characteristics (race, gender, age) to healthcare-related features ('time_in_hospital,' 'num_lab_procedures,' 'num_medications,' 'number_outpatient,' 'number_emergency,' 'number_inpatient,' 'number_diagnoses'). The class label 'readmitted' acts as a binary indicator, critical in assessing whether a patient was readmitted to the hospital within a specific timeframe after the initial admission. In this research, a subset of 2,001 instances from the dataset was utilized to develop predictive models aimed at efficiently forecasting patient readmissions. These models contribute to focusing on pressing challenges in healthcare, such as improving patient care outcomes and improving resource allocation. The selected subset enables efficient experimentation and model evaluation while ensuring that the results remain relevant and informative within the broader context of healthcare analytics [32].

	Label	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9
2	non-malicious	1	0	1	0	1	0	1	0	1
3	non-malicious	1	0	1	0	1	0	1	0	1
4	non-malicious	1	0	1	0	1	0	1	0	1
5	non-malicious	1	0	1	0	1	0	1	0	1
6	non-malicious	1	0	1	0	1	0	1	0	1
7	non-malicious	1	0	1	0	1	0	1	0	1
8	non-malicious	1	0	1	0	1	0	1	0	1
9	non-malicious	1	0	1	0	1	0	1	0	1
10	non-malicious	1	0	1	0	1	0	1	0	1
11	non-malicious	1	0	1	0	1	0	1	0	1
12	non-malicious	1	0	1	0	1	0	1	0	1
13	non-malicious	1	0	1	0	1	0	1	0	1
14	non-malicious	1	0	1	0	1	0	1	0	1
15	non-malicious	1	0	0	1	0	0	0	0	0
16	non-malicious	1	0	1	0	1	0	1	0	1
17	non-malicious	1	0	1	0	1	0	1	0	1
18	non-malicious	1	0	1	0	1	0	1	0	1
19	non-malicious	1	0	1	0	1	0	1	0	1
20	non-malicious	1	0	1	0	1	0	1	0	1
21	non-malicious	1	0	1	0	1	0	1	0	1
22	non-malicious	1	0	1	0	1	0	1	0	1
23	non-malicious	1	0	1	0	1	0	1	0	1

Figure 2. Malware Detection Dataset Sample

	race	gender	age	admissior	discharge	admissior	time_in_h	num_lab_	num_proc	num_med	number_c	number_e	number_i	number_c	readmitted
2	2	1	3	1	1	7	2	44	1	16	0	0	0	7	0
3	2	1	4	1	1	7	1	51	0	8	0	0	0	5	0
4	2	0	8	2	1	4	13	68	2	28	0	0	0	8	0
5	2	0	9	3	3	4	12	33	3	18	0	0	0	8	0
6	0	1	6	2	1	4	7	62	0	11	0	0	0	7	1
7	2	0	4	1	3	7	7	60	0	15	0	1	0	8	1
8	2	1	8	1	6	7	10	55	1	31	0	0	0	8	0
9	0	1	6	1	3	7	12	75	5	13	0	0	0	9	0
10	0	1	5	1	1	7	4	45	4	17	0	0	0	8	1
11	2	0	5	1	1	7	3	29	0	11	0	0	0	3	0
12	2	1	7	3	6	2	6	42	2	23	0	0	0	8	0
13	2	0	5	2	1	4	2	66	1	19	0	0	0	7	0
14	2	1	6	2	1	4	2	36	2	11	0	0	0	6	0
15	0	0	7	2	1	4	2	47	0	12	0	0	0	8	0
16	0	0	7	3	1	2	3	19	4	18	0	0	0	6	0
17	4	0	5	1	1	7	1	33	0	7	0	0	0	3	0
18	2	1	8	1	3	7	6	64	3	18	0	0	0	7	0
19	0	0	6	1	1	7	6	87	0	18	0	0	0	9	0
20	2	0	7	2	11	2	5	46	2	20	0	0	0	9	0
21	2	0	7	3	1	2	3	33	1	8	0	0	0	5	0
22	2	1	7	1	6	7	7	47	2	22	0	0	0	8	0
23	2	0	8	1	11	7	7	72	1	27	0	0	0	9	0

Figure 3. Hospital Readmission Dataset Sample

The Mushroom dataset, taken from the UCI Machine Learning Repository, is a comprehensive and broadly used resource for classification tasks related to mushroom edibility. This dataset includes a rich array of attributes that provide crucial understanding about mushroom characteristics, including cap shape, cap surface, cap color, bruises, odor, gill attachment, gill spacing, gill size, and gill color. With a total of 8,124 instances, the dataset provides a substantial sample size for analysis. Each attribute contributes significantly to determining whether a mushroom is edible or poisonous, making it an ideal choice for research in this domain [33].

We concentrated on a subset of the Mushroom dataset comprising 3,001 instances and nine key attributes: cap shape, cap surface, cap color, bruises, odor, gill attachment, gill spacing, gill size, and gill color. This subset worked as the foundation for our investigation into the classification of mushroom edibility based on these specific attributes. Using this subset allowed us to simplify the analysis while maintaining the critical features required to differentiate between edible and poisonous mushrooms, supporting a more comprehensive understanding of fungal species classification.

1	class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachment	gill-spacing	gill-size	gill-color
2	1	3	3	0	0	7	2	0	1	0
3	0	3	3	9	0	0	2	0	0	0
4	0	0	3	8	0	1	2	0	0	1
5	1	3	2	8	0	7	2	0	1	1
6	0	3	3	3	1	6	2	1	0	0
7	0	3	2	9	0	0	2	0	0	1
8	0	0	3	8	0	0	2	0	0	4
9	0	0	2	8	0	1	2	0	0	1
10	1	3	2	8	0	7	2	0	1	7
11	0	0	3	9	0	0	2	0	0	4
12	0	3	2	9	0	1	2	0	0	4
13	0	3	2	9	0	0	2	0	0	1
14	0	0	3	9	0	0	2	0	0	10
15	1	3	2	8	0	7	2	0	1	0
16	0	3	0	0	1	6	2	1	0	1
17	0	5	0	3	1	6	2	0	1	0
18	0	3	0	8	1	6	2	1	0	0
19	1	3	3	0	0	7	2	0	1	1
20	1	3	2	8	0	7	2	0	1	1
21	1	3	3	0	0	7	2	0	1	0
22	0	0	3	9	0	0	2	0	0	0
23	1	3	2	0	0	7	2	0	1	1

Figure 4. Mushroom Dataset Sample

Experiment and Results Discussion:

Three different models: DKNN (Dice Coefficient KNN), OKNN (Overlap Coefficient KNN), and SMKNN (Simple Match Coefficient KNN). These models are specifically designed for categorical data, and we have applied them to three diverse datasets: a malware detection dataset, the mushroom dataset, and a hospital readmission dataset. The main objective of our study is to evaluate the performance of these models across diverse domains and datasets, thereby assessing their applicability and effectiveness in different scenarios. Each of these models utilizes a K-nearest neighbors (KNN) approach, but they employ different similarity metrics, namely the Dice Coefficient, Overlap Coefficient, and Simple Match Coefficient, to calculate the similarity between data points. By comparing the performance of these models on each dataset, we can draw meaningful conclusions about their strengths and limitations. This analysis enables us to identify which model performs best for each dataset and under specific conditions. Such findings are essential for understanding the suitability of these models for real-world applications, as they provide awareness about the effectiveness of different similarity metrics in categorical data.

Case 1:

By comparing the performance of these models on each dataset, we can draw meaningful conclusions about their strengths and limitations. This analysis enables us to identify which model performs best for each dataset and under specific conditions. DKNN achieved an accuracy of 53.23%, while both OKNN 76.88% and SMKNN delivered remarkably high accuracy of 85.48%. The significant performance gap between DKNN and the other two models, OKNN and SMKNN, underscores the crucial impact of the chosen similarity coefficient in K-Nearest Neighbors-based classification. In our case, the choice of the Dice Coefficient in DKNN may not have been optimal for this specific dataset, resulting in significantly lower accuracy. In contrast, OKNN and SMKNN, which utilize the Overlap Coefficient and Simple Match Coefficient, respectively, showed excellent accuracy levels of 76.88% and 85.48% indicating their suitability for the task of malware detection. These results

point out the importance of tailoring machine learning models to the specific characteristics of the dataset at hand. The choice of similarity metric can significantly impact the model's performance, and in this case, SMKNN has proven to be highly effective for malware detection, achieving an impressive accuracy rate that holds great potential for enhancing cybersecurity practices.

Table 2. Testing the Models on the Malware Detection Dataset

Model	Accuracy	Precision	Recall	F1-Score
DKNN	53.23%	77%	60%	67%
OKNN	76.88%	77%	100%	87%
SMKNN	**85.48%**	**84%**	**100%**	**91%**

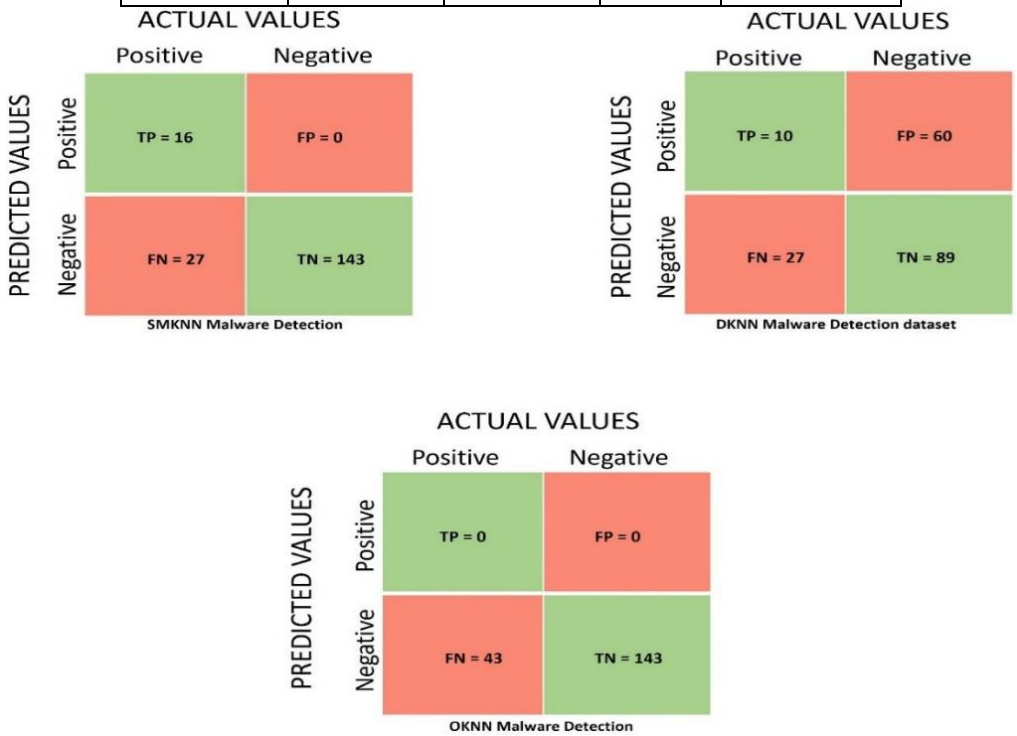

Figure 5. Malware Detection Dataset Confusion Matrix

Case 2:

In the context of the hospital readmission dataset, as shown in Table 4.3 below, the results of this investigation revealed valuable insights into the models' performance on a challenging healthcare-related task. DKNN achieved an accuracy of 79.00%, while OKNN and SMKNN delivered competitive accuracies of 81.60% and 80.10%, respectively. These results show that all three models, DKNN, OKNN, and SMKNN, performed comparably well on the hospital readmission dataset, achieving accuracy rates ranging from the high 70s to the low 80s. This indicates that, in this healthcare context, the choice of similarity coefficient, Dice for DKNN, Overlap for OKNN, and Simple Match for SMKNN, had minimal impact on the models' overall performance. Such consistent accuracy levels across the models underscore their potential utility in predicting patient readmission, a task of great importance in healthcare management and resource allocation. Further exploration and fine-tuning of these models may yield even more accurate and reliable readmission prediction systems, ultimately enhancing patient care and healthcare cost optimization.

Table 3. Testing the Models on the Hospital Readmission Dataset

Model	Accuracy	Precision	Recall	F1-Score
DKNN	79.00%	81%	97%	88%
OKNN	**81.60%**	**83%**	**97%**	**90%**

SMKNN	80.10%	84%	94%	89%

Case 3:

The outcomes of this analysis show interesting intuitions about the models' performance when tasked with distinguishing between edible and poisonous mushrooms. DKNN has achieved a respectable accuracy of 89.40%, while OKNN shows even higher accuracy at 93.20%. Especially, SMKNN stood out with an outstanding accuracy of 99.30%. These results show that all three models, DKNN, OKNN, and SMKNN, performed well on the mushroom dataset. However, SMKNN's outstanding accuracy of 99.30% suggests that the choice of the Simple Match Coefficient as a similarity metric is particularly well-suited for this specific classification task. The high accuracy rate of SMKNN draws attention to its effectiveness in reliably identifying edible and poisonous mushrooms, which has significant implications for food safety and mycological research. These findings put emphasis on the importance of selecting the appropriate model and similarity metric for the unique characteristics of the dataset, as this choice can greatly impact the accuracy and trustworthiness of the classification results.

Table 4. Testing the Models on the Mushrooms Dataset

Model	Accuracy	Precision	Recall	F1-Score
DKNN	89.40%	89%	100%	94%
OKNN	93.20%	94%	99%	96%
SMKNN	**99.30%**	**100%**	**100%**	**100%**

Average Results:

The result of the whole experiment shows an interesting knowledge about the performance of the three categorical data models SMKNN, OKNN, and DKNN across the selected datasets. Among these models, SMKNN presented the highest average accuracy, achieving an impressive accuracy rate of 88.29%. This outcome suggests that the Simple Match Coefficient, which evaluates the proportion of matching attribute values, is especially effective at capturing the underlying patterns and relationships within the dataset. Trailing slightly, OKNN recorded an average accuracy of 83.89%, highlighting its effectiveness in handling categorical data. The Overlap Coefficient used in OKNN, which focuses on the intersection of attribute values, proved to be a worthy similarity metric for these datasets. Conversely, DKNN lagged with the lowest average accuracy of 73.74%. This result suggests that the Dice Coefficient, which calculates the ratio of shared attribute values to total attribute values, may be less suitable for these datasets, or that the selected value of K in DKNN failed to adequately capture the local patterns in the data. The observed variations in performance highlight the importance of selecting an appropriate similarity metric and tuning the model parameters to suit the characteristics of the dataset. It is also worth considering the computational complexity of each model, as this can impact its practical utility in diverse applications. The results of this experiment underline the significance of model selection and parameter tuning in categorical data analysis. SMKNN and OKNN demonstrated strong capability for accurate classification, while DKNN's relatively lower performance suggests the need for further refinement or exploration of other techniques. These findings provide valuable guidance for practitioners and researchers within the domain of machine learning, emphasizing the need to associate model choices with the specific requirements of the dataset and application domain.

Table 5. The Average Results of the Machine Learning Models

Model	Average Accuracy	Average Precision	Average Recall	Average F1-Score
DKNN	73.74%	82.33%	85.66%	83%
OKNN	83.89%	84.66%	98.66%	91%
SMKNN	**88.29%**	**89.33%**	**98%**	**93.3%**

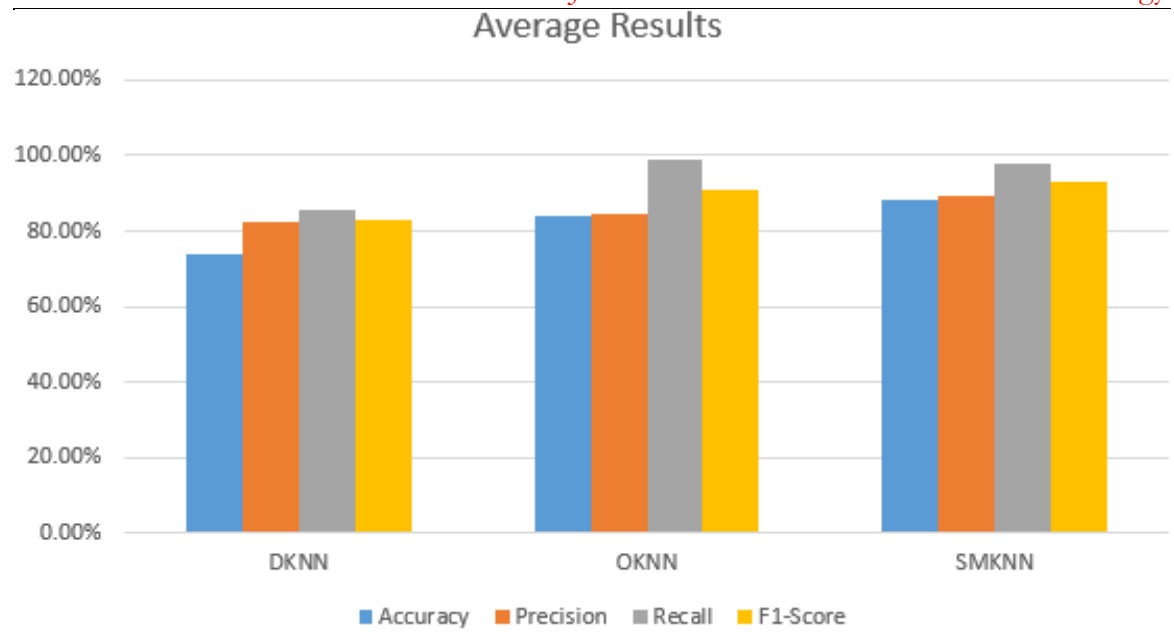

Figure 6. Average Results Graph

Conclusion:

The study has explored the performance of three K-Nearest Neighbors (KNN) variants, DKNN, OKNN, and SMKNN, with three different categorical datasets: a malware detection dataset, a mushroom dataset, and a hospital readmission dataset. Results show that the choice of similarity coefficient plays a significant influence in model performance. OKNN and SMKNN consistently give us high accuracy in many datasets, showing their effectiveness for different classification tasks. We have investigated and compared the performance of three distinctive K-Nearest Neighbors (KNN) models, DKNN, OKNN, and SMKNN, across various categorical datasets, including a malware detection dataset from Kaggle, a mushroom dataset from UCI, and a hospital readmission dataset from Kaggle [33]. The Results of our analysis point out the important role of the choice of similarity coefficient in KNN-based classification. OKNN and SMKNN consistently provide us with strong and competitive accuracy levels, showing their flexibility to various classification tasks. SMKNN's exceptional performance in recognizing edible and poisonous mushrooms, achieving an accuracy of 99.30%, stands out as a notable highlight. These findings emphasize the importance of selecting an appropriate model and a similar metric tailored to the dataset's unique characteristics. Overall, our study contributes valuable insights into the suitability of these KNN variants for specific domains, highlighting their potential for enhancing cybersecurity, mycology, and healthcare applications. Further research and refinement of these models hold promise for advancing classification tasks in these fields.

Future Work:

Exploration of Additional Similarity Measures: Extend your research by investigating a broader spectrum of similarity measures specifically designed for categorical data. This exploration can encompass Hamming distance, Sørensen-Dice coefficient, and other specialized metrics to gauge their impact on model performance.

Hybrid Similarity Combinations: Explore hybrid similarity combinations by combining or weighting multiple similarity measures. Investigate how combinations of these measures can be optimized to enhance model accuracy and robustness across various datasets.

Imbalanced Data Handling: Investigate techniques for handling imbalanced categorical datasets, such as oversampling, undersampling, or using specialized cost-sensitive learning approaches.

Multi-class Classification: Extend your research to multi-class classification scenarios, examining how well KNN models with categorical data can handle multiple classes and categories.

Real-time Processing: Adapt your models to real-time or streaming data applications, addressing challenges associated with continuous data updates and dynamic environments.

Data Availability Statement: Data will be made available on request.

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

Ethical Approval and Consent to Participate: Not applicable.

References:

[1] V. K. Varun Chandola, Shyam Boriah, "A Framework for Exploring Categorical Data," *Proc. 2009 SIAM Int. Conf. Data Min.*, pp. 187–198, 2009, doi: https://doi.org/10.1137/1.9781611972795.17.

[2] Roy Thomas, "A Novel Ensemble Method for Detecting Outliers in Categorical Data," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 4, pp. 4947–4953, 2021, doi: 10.30534/ijatcse/2020/108942020.

[3] Mohammad Bolandraftar, "Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background," *Academia*, 2013, [Online]. Available: https://www.academia.edu/82354140/Application_of_K_Nearest_Neighbor_KNN_Approach_for_Predicting_Economic_Events_Theoretical_Background

[4] O. L. V. B. Surya Prasath, Haneen Arafat Abu Alfeilat, Ahmad B. A. Hassanat, "Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier -- A Review," *Big Data*, vol. 7, no. 4, pp. 221–248, 2019, doi: https://doi.org/10.48550/arXiv.1708.04321.

[5] B. Guindon and Y. Zhang, "Application of the Dice Coefficient to Accuracy Assessment of Object-Based Image Classification," *Can. J. Remote Sens.*, vol. 43, no. 1, pp. 48–61, Jan. 2017, doi: 10.1080/07038992.2017.1259557.

[6] Q. Zheng, "From Whole to Part: Reference-Based Representation for Clustering Categorical Data," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 3, pp. 927–37, 2020.

[7] Z. C. Semeh Ben Salem, Sami Naouali, "A fast and effective partitional clustering algorithm for large categorical datasets using a k-means based approach," *Comput. Electr. Eng.*, vol. 68, pp. 463–483, 2018, doi: https://doi.org/10.1016/j.compeleceng.2018.04.023.

[8] V. T. Jaglan, Vivek, "Comparison of Jaccard, Dice, Cosine Similarity Coefficient To Find Best Fitness Value for Web Retrieved Documents Using Genetic Algorithm," *Int. J. Innov. Eng. Technol.*, vol. 2, no. 4, pp. 202–205, 2013, [Online]. Available: https://www.researchgate.net/publication/306204167_Comparison_of_Jaccard_Dice_Cosine_Similarity_Coefficient_To_Find_Best_Fitness_Value_for_Web_Retrieved_Documents_Using_Genetic_Algorithm

[9] M. Tay and A. Senturk, "A New Energy-Aware Cluster Head Selection Algorithm for Wireless Sensor Networks," *Wirel. Pers. Commun.*, vol. 122, no. 3, pp. 2235–2251, Feb. 2022, doi: 10.1007/S11277-021-08990-3/METRICS.

[10] L. D. Amir Ahmad, "A k-means type clustering algorithm for subspace clustering of mixed numeric and categorical datasets," *Pattern Recognit. Lett.*, vol. 32, no. 7, pp. 1062–1069, 2011, doi: https://doi.org/10.1016/j.patrec.2011.02.017.

[11] R. S. R. Minho Kim, "Projected clustering for categorical datasets," *Pattern Recognit. Lett.*, vol. 27, no. 12, pp. 1405–1417, 2006, doi: https://doi.org/10.1016/j.patrec.2006.01.011.

[12] T. S. Mohammed J. Zaki, Markus Peters, Ira Assent, "Clicks: An effective algorithm

for mining subspace clusters in categorical datasets," *Data Knowl. Eng.*, vol. 60, no. 1, pp. 51–70, 2007, doi: https://doi.org/10.1016/j.datak.2006.01.005.

[13] S. Sharma and M. Singh, "Generalized similarity measure for categorical data clustering," *2016 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2016*, pp. 765–769, Nov. 2016, doi: 10.1109/ICACCI.2016.7732138.

[14] H. L. Chen, K. T. Chuang, and M. S. Chen, "On data labeling for clustering categorical data," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 11, pp. 1458–1471, Nov. 2008, doi: 10.1109/TKDE.2008.81.

[15] D. T. Nguyen, L. Chen, and C. K. Chan, "Clustering with multiviewpoint-based similarity measure," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 6, pp. 988–1001, 2012, doi: 10.1109/TKDE.2011.86.

[16] S. Mumtaz and M. Giese, "Frequency-Based vs. Knowledge-Based Similarity Measures for Categorical Data," *AAAI Spring Symp. Comb. Mach. Learn. with Knowl. Eng.*, 2020.

[17] X. Zeng, Y. Liao, Y. Liu, and Q. Zou, "Prediction and validation of disease genes using HeteSim scores," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 14, no. 3, pp. 687–695, May 2017, doi: 10.1109/TCBB.2016.2520947.

[18] R. Devika, S. V. Avilala, and V. Subramaniyaswamy, "Comparative study of classifier for chronic kidney disease prediction using naive bayes, KNN and random forest," *Proc. 3rd Int. Conf. Comput. Methodol. Commun. ICCMC 2019*, pp. 679–684, Mar. 2019, doi: 10.1109/ICCMC.2019.8819654.

[19] H. C. Sayali D. Jadhav, "Comparative Study of K-NN , Naive Bayes and Decision Tree Classification Techniques," *Int. J. Sci. Res.*, vol. 5, no. 1, pp. 1842–1845, 2016, [Online]. Available: https://www.semanticscholar.org/paper/Comparative-Study-of-K-NN-%2C-Naive-Bayes-and-Tree-Jadhav-Channe/51c068c263ee197a292df5b74b58c8c55df9f9ca

[20] Kedar Potdar, Taher S. Pardawala, Chinmay D. Pai, "A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers," *Int. J. Comput. Appl.*, vol. 175, no. 4, 2017, [Online]. Available: https://www.ijcaonline.org/archives/volume175/number4/potdar-2017-ijca-915495.pdf

[21] K. Das, J. Schneider, and D. B. Neill, "Anomaly pattern detection in categorical datasets," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 169–176, 2008, doi: 10.1145/1401890.1401915.

[22] B. R. Parida and S. P. Mandal, "Polarimetric decomposition methods for LULC mapping using ALOS L-band PolSAR data in Western parts of Mizoram, Northeast India," *SN Appl. Sci.*, vol. 2, no. 6, pp. 1–15, 2020, doi: 10.1007/s42452-020-2866-1.

[23] A. A. Nair, T. D. Tran, A. Reiter, and M. A. Lediju Bell, "A Deep Learning Based Alternative to Beamforming Ultrasound Images," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2018-April, pp. 3359–3363, Sep. 2018, doi: 10.1109/ICASSP.2018.8461575.

[24] M. C. Rahim Taheri, Meysam Ghahramani, Reza Javidan, Mohammad Shojafar, Zahra Pooranian, "Similarity-based Android malware detection using Hamming distance of static binary features," *Futur. Gener. Comput. Syst.*, vol. 105, pp. 230–247, 2020, doi: https://doi.org/10.1016/j.future.2019.11.034.

[25] L. E. Tiago R.L. dos Santos, Zárate, "Categorical data clustering: What similarity measure to recommend?," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1247–1260, 2015, doi: https://doi.org/10.1016/j.eswa.2014.09.012.

[26] C. W. Yean *et al.*, "Analysis of the Distance Metrics of KNN Classifier for EEG Signal in Stroke Patients," *2018 Int. Conf. Comput. Approach Smart Syst. Des. Appl.*

ICASSDA 2018, Sep. 2018, doi: 10.1109/ICASSDA.2018.8477601.

[27] S. Naseer, "Enhanced Network Anomaly Detection Based on Deep Neural Networks," *IEEE Access*, vol. 6, pp. 48231–48246, 2018, doi: 10.1109/ACCESS.2018.2863036.

[28] R. K. Amin, Indwiarti, and Y. Sibaroni, "Implementation of decision tree using C4.5 algorithm in decision making of loan application by debtor (Case study: Bank pasar of Yogyakarta Special Region)," *2015 3rd Int. Conf. Inf. Commun. Technol. ICoICT 2015*, pp. 75–80, Aug. 2015, doi: 10.1109/ICOICT.2015.7231400.

[29] V. S. Tseng and C. P. Kao, "Efficiently mining gene expression data via a novel parameterless clustering method," *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 2, no. 4, pp. 355–365, Oct. 2005, doi: 10.1109/TCBB.2005.56.

[30] C. D. Cao, Fuyuan, Jiye Lian, Deyu Li, Liang Bai, "A dissimilarity measure for the k-Modes clustering algorithm," *Knowledge-Based Syst.*, vol. 26, pp. 120–127, 2012, doi: https://doi.org/10.1016/j.knosys.2011.07.011.

[31] V. Minkevics and J. Kampars, "Methods, models and techniques to improve information system's security in large organizations," *ICEIS 2020 - Proc. 22nd Int. Conf. Enterp. Inf. Syst.*, vol. 1, pp. 632–639, 2020, doi: 10.5220/0009572406320639.

[32] H. N. Pham *et al.*, "Predicting hospital readmission patterns of diabetic patients using ensemble model and cluster analysis," *Proc. 2019 Int. Conf. Syst. Sci. Eng. ICSSE 2019*, pp. 273–278, Jul. 2019, doi: 10.1109/ICSSE.2019.8823441.

[33] M. M. Eyad Alkronz, Khaled A. Moghayer, "Prediction of Whether Mushroom is Edible or Poisonous Using Back-propagation Neural Network.," *Int. J. Corpus Linguist.*, vol. 3, no. 2, pp. 1–8, 2019, [Online]. Available: https://www.researchgate.net/publication/331464780_Prediction_of_Whether_Mushroom_is_Edible_or_Poisonous_Using_Back-propagation_Neural_Network