University of Ibadan
Journal of Science and Logics in ICT Research

Index Mapped Ordinal Encoding Method for Federated Machine Learning in Crime Detection

AYINLA, B. I.

Department of Computer Science, University of Ibadan, Nigeria
Corresponding Author: ayinlab@gmail.com

Abstract

Feature Engineering is extracting features from a raw dataset using the understanding of the domain's problem. Thus, an improved feature extraction process can enhance the performance of machine learning algorithms. The resultant effect is the increase in the accuracy of a model in detecting new knowledge. However, the pyramid of unstructured data generated by surveillance devices and business transactions into databases is alarming and calls for serious attention. Transforming this messy data to a machine-useable format at the edge for prediction and classification exercises is therefore challenging. Although there have been some techniques for this data pre-processing phase in the model building but they are not totally spared from loss of data, duplication and difficult implementation. This is an examination of the efficacy of a novel Index Mapped Ordinal Encoding Method (IMOEM) for machine learning algorithm in terms of precision, recall and accuracy. The performance of IMOEM built on this dataset with respect to precision, recall, f-score and accuracy results were significantly effective. The model performed exceptionally well with no loss of accuracy either in precision or recall values, especially when applied to the decision tree based Models. Data scientists are therefore encouraged to embrace the use of IMOEM.

Keywords: Encoding Method, Knowledge Extraction, Data Accuracy, Feature Engineering, Federated Machine Learning Algorithm

1.0. Introduction

Artificial Intelligence (AI) has been a buzzword for quite some time and is highly ubiquitous. AI-enabled applications have extensively increased in the market. There are also powerful infrastructures and advanced algorithms to analyse the pyramid of data available in the data space today. However, this has not made the acquisition of insightful knowledge of Machine Learning (ML) projects easier.

According to Sikibi [27], it is evident that data is essential to the survival of every organisation. Organisational reports, whether daily, weekly, monthly or even yearly, can offer information on how a business operates and performs. The client data might reveal details about the intended market and its commercial tendencies. Transactional data allows an organisation to raise brand awareness and increase profit [22].

Similarly, information on customers, purchases, expenses and workers can enable organisations to make well-informed decisions that can advance the business horizon. Although this sounds good, analysing and processing such unstructured data can be a difficult undertaking and, if not done effectively, could prevent the organisation from realising the benefits mentioned earlier [27].

Cremer [10] asserted that organisations and cybersecurity specialists must comb through a sizeable amount of business data produced by various devices and apps such as network logs, email logs, access logs and databases in order to learn from them. The report further re-echoes the fact that unstructured data is becoming increasingly significant to an organisation’s success. According to a recent forecasts, unstructured data makes up more than 80% of all enterprise data, and 95% of firms prioritised managing this data (IBM Cloud Education, 2021).

Ayinla, B. I. (2023). Index Mapped Ordinal Encoding Method for Federated Machine Learning in Crime Detection,.*University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)*, Vol. 9 No. 1, pp. 1-15

It is necessary to transform the raw data into clear and practical information to make predictions. In order to create useable datasets for cybersecurity detection and prediction, data need to be transformed into a form that can be used for pattern mining.

Furthermore, feature engineering is a major area of data extraction that deals with raw datasets by utilising domain knowledge. In line with the submission of Patel [16], an improved feature extraction process improves the performance of machine learning algorithms by increasing the accuracy of a model's prediction and detection. It is the most critical part of machine learning and contributes significantly to distinguishing between good and bad models. A good feature selection process will result in the designing of efficient machine learning models. Supposing the real-world events are to be represented symbolically; according to [11] and [17], data validity for both model construction and prediction must receive enough attention.

More importantly, efforts must be geared towards translating raw observations into a valuable form for detecting criminal activities or organisation progress ([9]; [26]. A large portion of the data produced by security devices are presently deficient. They require tools to analyse or convert them into a format that is appropriate for relevant pattern extraction on how criminals strategically intend to carry out attacks [11].

Moreover, data transformation and cleaning is often simultaneously constructive and destructive and it is the process of removing redundant features, attributes, and instances. Missing variables may be added to the data to make it easier to detect hidden insights. It may be essential to rename some fields and combine many columns into one at this stage [31]. The quality and dependability of a model created from the information for harmful detection would be significantly improved when data cleaning is carried out with much diligence [18,5] and [2].

Machine learning techniques are widely used in cybersecurity in a variety of ways. Modern cybersecurity can now perform malware detection, intrusion detection, and data leakage which were previously impossible to carry out using only mathematical models [4]. In today's competitive world, every organisation requires a well-designed and long-term data strategy to deal with the obvious complexities of multi-source, multi-type, and very high volumes of data pouring in from the most recent technological funnels [7]. Similarly, the Internet of Things (IoT) ushers in the new era of the Industry 4.0 revolution. It is a global infrastructure for gathering and processing vital and sensitive data from our surroundings for storage, actuation, sensing, advanced services and communication technologies. However, it comes with its own challenges such as security risks and missing values, which could occur during data transformation.

According to Peter [24], the security breaches of Facebook users’ data in 2018 spawned the use of Artificial Intelligence (AI) in cybercrime. As a result, organisations are now tasked not to betray the public trust of their teaming customers against the advances of cybercriminals. They must prioritise the use of AI tools to protect their customers' data.

The rest of this paper is organised as follows. The next section is the literature review on data transformational methods: One-hot Encoding or Dummies Encoding, Map Ordinal Values to Numbers and related cybersecurity literature. Section 3 describes the proposed method of encoding: Index Map Ordinal Method. In Section 4, the experimental results are discussed. Finally, Section 5 provides the limitations, recommendations and some conclusions.

2.0. Related Works

It is well known that categorical observations have the ability to conceal and mask knowledge in a dataset [2][8][18]. Learning how to deal with such aspects is rather a crucial step. There are essentially just two widely used data encoding methods, according to Brownlee [17], One-Hot Encoding or Dummies Encoding and Map Ordinal Values
to Numbers.

2.1. One-Hot Encoding or Dummies Encoding

When building a detective system, there are several ways to improve Machine Learning Models (MLMs), especially by transforming the dataset into machine useable format. One example of these strategies is the One-hot or Dummy Encoding which assigns a numerical value to categorical features[6] and [14]. Due to the fact that many machine learning algorithms cannot work with text variables, this dataset must be converted to numerical values116]. The target variables may need to be presented or used in some applications in a categorical form at some points [16] & [18].

This method is a straightforward non-parametric approach that may be used for any kind of categorical variable without making any assumptions about the values of the variables, according to [18]. Take a dataset with five unique features as an illustration. Each of the category features in the original dataset is represented by one of five different numerical values.

However, it only supports up to 15 classified variables and it performs flawlessly [18]. This data transformation technique does not require categorical data as an input. It is, therefore, best used as a grounding strategy that we can fall back on if no other technique can carry out the task more efficiently and intuitively. The get dummies method in the Pandas Python Library provides support for this transition [31][20].

To demonstrate this method, let each of the features in Table 1 be mapped to a vector containing 1 and 0 showing the feature's presence or absence. The number of vectors depends on the number of categories of features. Tables (1) and (2) show an example of the One-Hot Encoding Transformation.

2.2. Map Ordinal Encoding Method (MOEM)

This is the most basic encoding technique in which each value in the categorical dataset is converted to an integer value. The values are assigned to each distinct feature in no particular order other than the order in which the columns appeared but this can be changed using the order argument to the function [6][28].

Because the allocation of unique numbers frequently begins with 0, the data is transformed from a k-class ordinal state to a k-1 integer number of variables

Table 1: Categorical Raw Programming Training Dataset

	Programming Training	Class	City
0	Python	Interpreter	Lagos
1	Pascal	Compiler	Ibadan
2	Fortran	Compiler	Ibadan
3	Pascal	Hybrid	Lagos
4	Fortran	Interpreter	Ibadan
5	Python	Interpreter	Lagos
6	Pascal	Compiler	Lagos

Table 2: One-hot Encoding Format of Programming Training Dataset

	Python	Pascal	Fortran	Class	City_1	City_2
0	1	0	0	Interpreter	1	0
1	0	1	0	Compiler	0	1
2	1	0	1	Compiler	0	1
3	0	1	0	Hybrid	1	0
4	0	0	1	Interpreter	0	1
5	1	0	0	Interpreter	1	0
6	0	1	0	Compiler	1	0

Brownlee [18] re-aligned what is an ordinal technique of transforming categorical features for the underlying machine learning algorithm to the numerical format in the article, "Ordinal and One-hot Encodings for Categorical Data" published on the internet. Encoding begins with converting each distinct categorical value to an integer value[28]. It is, however, easily reversible. An ordinal encoding may be sufficient for some variables[16] and [21]. Natural integer values have an intrinsic ordered relationship with one another which machine learning algorithms can understand and exploit

[14]. For example, crime categories in Baltimore city, "Robbery", "Larceny" and "Assault", are converted into integers, as illustrated in Tables 3 and 4. First, the categories are sorted then numbers are applied. For strings, this means the labels are sorted alphabetically as Assault=0, Larceny=1 and Robbery=2 [18][24].

Table 3: Categorical Crime Dataset

	Crime
0	Assault
1	Larceny
2	Robbery
3	Assault
4	Larceny
5	Robbery
6	Assault
7	Larceny
8	Robbery

Table 4: Ordinal Encoding Format of Crime Dataset

	Height	Transformed
0	Assault	0
1	Larceny	1
2	Robbery	2
3	Assault	0
4	Larceny	1
5	Robbery	2
6	Assault	0
7	Larceny	1
8	Robbery	2

The Scikit-learn Python machine learning has a function in its library known as the OrdinalEncoder class to perform raw data ordinal encoding. By default, it assigns integer values in the manner in which the data appeared based on the underlying order.

2.3. Related Cybersecurity Literature

Reviewing the effects of Artificial Intelligent (AI) approaches deployed between 2016 and 2021, Alloghani [3] show that AI was a leading cybersecurity technique before 2016. Also, a lot of research is currently focusing on using evolutionary algorithms, fuzzy logic and neural networks to detect cybersecurity threats. Intrusion Detection Systems (IDSs) with an embedded intelligence architecture framework was used to alleviate the negative effect of cyber crimes. In 2015, Dilek [13] publish a thorough analysis of the application of AI to combat cybercrime, including the effectiveness of the techniques to identify and thwart cyberattacks.

Alsoufi [5] [21] assert the implementation of Machine Learning (ML) and Data Mining (DM) to improve cybersecurity and limit vulnerabilities to cyber attacks. The article reiterates that most attacks succeeded because their victims lacked the tools and knowledge to defend themselves effectively. Despite these difficulties, machine learning models have been really effective as tools for mining the operational patterns of cybercriminals. Moreso, with the proliferation of IoT devices or smart devices in homes and offices, there has been a recent surge of global interests in cyber security [5]. These devices are handy to the point that criminals can use them to conduct cyber-attacks at any time and from any location. According to a cybersecurity firm, Helsinki, a Finland-based F-Secure, based on Gartner's in 2020 research, the number of Internet of Things (IoT) devices per household will skyrocket to 500 by 2022 from nine currently, with IoT connectivity included in the package, without being optional [7] and [23].

Mikko Hypponen, the Chief Research Officer for F-Secure, published a report on IoT devices in 2018 stating that devices lacking IoT capabilities may no longer be affordable because manufacturers may be unable to harvest data from them, whereas it is this data that makes the IoT ideal for businesses. However, the data come with a variety of challenges or risk factors that need to be addressed quickly. Among these challenges are, but are not limited to, security and trust, reliability, scalability and mobility [3][7].

Sarker [25] respond to cyber threats by examining the ranking of data security features using artificial intelligence tools, specifically Machine Learning (ML) techniques. To extract data patterns, an intrusion detection system was adopted that used a machine-learning-based security model known as Intrusion Detection Tree ("IntruDTree"). It took into account the importance of security features when ranking them. Based on the significance of the selected features, a tree-based generalised intrusion detection model was constructed. When used on the test set, the model had improved prediction accuracy and was effective in terms of computational complexity due to the reduction in the number of features chosen. The model was tested on cybersecurity datasets, and its performance was evaluated using precision, recall, f-score, accuracy, and ROC values. To assess the effectiveness of the machine learning security model, the IntruDTree model was compared to some traditional popular machine learning models such as the Naive Bayes (NB), Logistic Regression (LR), Support Vector Machines (SVMs), and K-Nnearest Neighbor (KNN). According to Alshaibi [4], most Industrial Internet of Things (IIoT) attacks occur at the data transmission layer, as reported by many sources. In the industrial internet of things, Intrusion Detection System (IDS) models are now based on Machine Learning (ML) and Deep Learning (DL) techniques to detect malicious patterns in any layer of its architecture. The nature of datasets used in training the IDS model was the primary focus of exploration and analysis of major

IoT attacks. The review emphasised that tool hybridisation is more effective than stand-alone machine learning techniques. It is, therefore, sufficient to say that where two or more artificial intelligence instruments collaborate, there is improvement in tracking cyber attacks.

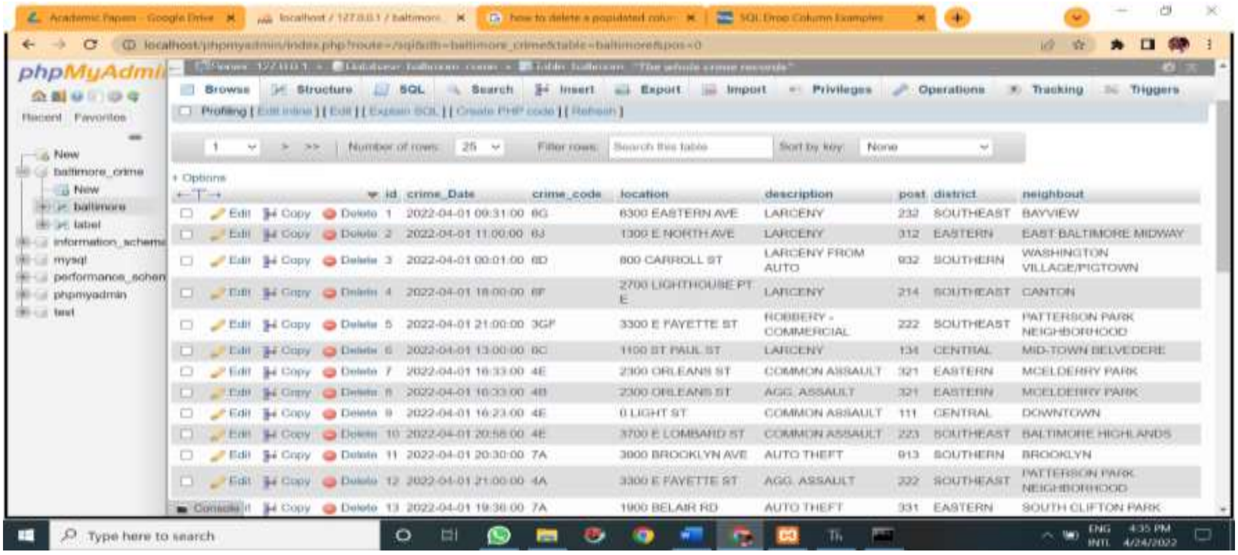
In a similar manner, Abdullahi [1] reiterated in the article titled "Detecting Cybersecurity Attacks in the Internet of Things Using Artificial Intelligence Methods: A Systematic Literature Review". The report submitted that Machine Learning (ML) as an instrument of intrusion detection is perfect. The literature considered the volume of data generated via the Internet of Things (IoT) devices and networks in various forms which required careful authentication and security. The method is a promising technique for addressing cybersecurity threats and providing adequate security measures in an organisation. However, many studies have introduced smart Intrusion Detection Systems (IDS), with intelligent architectural frameworks on smart devices, to lessen the success of cybercriminals.

3.0. Methodology

The four machine learning models used are Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbour (KNN) and Decision Tree (DT). These models were constructed using Python programming language on Jupyter Lab package housed by the Windows Operating System.

3.1. Data collection

Four machine learning models were built using the data transformation methods introduced in this research. Consequently, 276, 534 raw instances of Baltimore crime categorical dataset were used. The dataset had 15 features as depicted in Figure 1.



id	crime_Date	crime_code	location	description	post	district	neighbout
1	2022-04-01 09:31:00	6G	6300 EASTERN AVE	LARCENY	232	SOUTHEAST	BAYVIEW
2	2022-04-01 11:00:00	6J	1300 E NORTH AVE	LARCENY	312	EASTERN	EAST BALTIMORE MIDWAY
3	2022-04-01 00:01:00	6D	800 CARROLL ST	LARCENY FROM AUTO	932	SOUTHERN	WASHINGTON VILLAGE/PIGTOWN
4	2022-04-01 18:00:00	6F	2700 LIGHTHOUSE PT E	LARCENY	214	SOUTHEAST	CANTON
5	2022-04-01 21:00:00	3GF	3300 E FAYETTE ST	ROBBERY - COMMERCIAL	222	SOUTHEAST	PATTERSON PARK NEIGHBORHOOD
6	2022-04-01 13:00:00	6C	1100 ST PAUL ST	LARCENY	134	CENTRAL	MID-TOWN BELVEDERE
7	2022-04-01 16:33:00	4E	2300 ORLEANS ST	COMMON ASSAULT	321	EASTERN	MCELDERRY PARK
8	2022-04-01 16:33:00	4B	2300 ORLEANS ST	AGG. ASSAULT	321	EASTERN	MCELDERRY PARK
9	2022-04-01 16:23:00	4E	0 LIGHT ST	COMMON ASSAULT	111	CENTRAL	DOWNTOWN
10	2022-04-01 20:58:00	4E	3700 E LOMBARD ST	COMMON ASSAULT	223	SOUTHEAST	BALTIMORE HIGHLANDS
11	2022-04-01 20:30:00	7A	3000 BROOKLYN AVE	AUTO THEFT	913	SOUTHERN	BROOKLYN
12	2022-04-01 21:00:00	4A	3300 E FAYETTE ST	AGG. ASSAULT	322	SOUTHEAST	PATTERSON PARK NEIGHBORHOOD
13	2022-04-01 19:36:00	7A	1900 BELAIR RD	AUTO THEFT	331	EASTERN	SOUTH CLIFTON PARK

Figure 1: The 15-Columns Baltimore_Crime Database

The number of total features from the raw dataset

1. LARCENY
2. LARCENY FROM AUTO
3. ROBBERY - COMMERCIA
4. COMMON ASSAULT
5. AGG. ASSAULT
6. AUTO THEFT
7. BURGLARY
8. ROBBERY - STREET
9. ROBBERY - RESIDENCE
10. ROBBERY - CARJACKING
11. SHOOTING
12. ARSON
13. RAPE
14. HOMICIDE
15. TOTAL INCIDENCE

3.2 Data Classification Process/Pattern Mining Techniques

The research model grouped the processing activities into four phases: data downloading and housing into a relational database, data cleaning and dimensionality reduction, data transformation and machine learning model building.

3.2.1 Data Downloading and Warehousing

The raw dataset of crime committed in Baltimore in the United State of America, between the years 2012 and 2017, was downloaded from the Kaggle website (https://www.kaggle.com/search?q=baltimore) .Hence, a database named Baltimore_crime was created from the comma-separated values file using Mysql relational database known as the best data mining relational database [20],[15] & [21]. As earlier mentioned, the database initially had a Baltimore table, which housed the raw dataset with 15 columns. The dataset was unorganised and full of redundancies which was not fit for data mining processes or pattern recognition. Consequently, the Structural Query Language (SQL) was introduced to pick out missing attributes and duplicated instances. This was relatively easy to do.

3.2.2. Missing Attributes or Removal of Incomplete Instances

The missing attributes were filled using the values of instances with similar other attributes. At the same time, some tuples, especially those with more than half of the attributes missing, were outrightly removed.

3.2.3. Dimensionality Reduction

The table, "Baltimore", was the main table in the database with initial 15 columns and 276, 534 instances as in Figure 1. However, some attributes in the columns were collapsed and the column (feature) was subsequently renamed to depict the actual representation of the data. Some composite attributes were also broken down for deep mining to aid the model's training. The features that were totally irrelevant to the research query were truncated to reduce the feature dimensionality in order to facilitate effective model training. The attributes like Robbery-Street, Robbery-Resident, Robbery-commercial and Robbery-Carjacking were classified as Robbery, Larceny and Larceny from Auto were also collapsed into Larceny. The Common Assault and Agg Assualt formed Assault features using the structural query language statements as in Figure 2.

```
UPDATE `baltimore` SET `label`='ASSAULT'
WHERE `label` LIKE '%ASSAULT%';
UPDATE `baltimore` SET
`label`='LARCENY' WHERE `label` LIKE
'%LARCENY%';
UPDATE `baltimore` SET
`label`='ROBBERY' WHERE `label` LIKE
'%ROBBERY%';

SELECT * FROM `baltimore` WHERE `label`
NOT LIKE 'LARCENY' AND `label` NOT
LIKE 'ROBBERY' AND `label` NOT LIKE
'ASSAULT';

DELETE FROM `baltimore` WHERE `label`
NOT LIKE 'LARCENY' AND `label` NOT
LIKE 'ROBBERY' AND `label` NOT LIKE
'ASSAULT';
```

Figure 2: Structural Query Language Statements To Rename and Eliminate some Instances

3.2.4 Transforming Features/Columns to Table

The Indexed Identifiers Encoding Method was done using the structural query language without moving the dataset outside its original table. At the end of the data cleaning and dimensionality reduction activities, 8 distinct features were left. They were crime_code, crime_date (the date when the crime was committed), crime_time (the particular time of the day when the crime was carried out), crime_district (in which jurisdiction was the crime committed). The remaining features were crime_location (the particular spot where the crime was committed), crime_neighbourhood (the closest spot to the crime's location). One of the features was selected as the target, that is, the label. This was done using the structural query language statements as illustrated in Fig. 3. These features were converted to tables with an automatic indexing of the attributes.

```
INSERT INTO crime_code (code)
SELECT DISTINCT(`crime_code`) FROM
`baltimore` WHERE 1;
INSERT INTO
`crime_neighbourhood`(neighbourhood)
SELECT DISTINCT(`neighbout`) FROM
`baltimore` WHERE 1;
SELECT date_format(`crime_Date`,
'%H:%i:%s') as 'time' FROM `baltimore`;
INSERT INTO crime_time
(crime_time,rec_id) SELECT
DISTINCT(date_format(`crime_Date`,
'%H:%i:%s')) as 'time',`id` FROM `baltimore`;
INSERT INTO crime_Date (crime_Date,
rec_id) SELECT DATE(`crime_Date`), id
FROM baltimore;
```

Figure 3: Query Statements to populate Tables of Feature with Unique Attributes

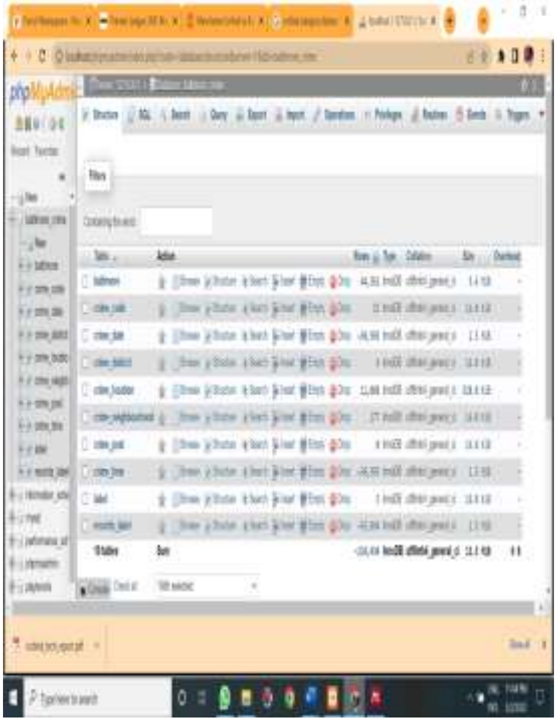


Figure 4: The 10-table Baltimore_crime database

3.2.5. Index Mapped Ordinal Data Discretization

The main table was restructured during data cleaning and the categorical attributes (values) in each column were replaced with their corresponding numeric index identifiers from the feature's tables, as shown in Figure 4. The final output of the Baltimore table contained only digital values, which were ready to be used to train the models. The transformations of the attributes into numerical values were done with the use of structural query language statements in Fig. 5.

```
UPDATE `baltimore`
   SET `baltimore`.`location` = (
   SELECT `crime_location`.id
   FROM `crime_location`
   WHERE `crime_location`.`location` =
`baltimore`.`location`
);

UPDATE `baltimore`
SET `baltimore`.crime_code = (
   SELECT `crime_code`.id
   FROM `crime_code`
   WHERE `crime_code`.code =
`baltimore`.`crime_code`
);
UPDATE `baltimore`
   SET `baltimore`.`district` = (
   SELECT `crime_district`.id
   FROM `crime_district`
   WHERE `crime_district`.`district` =
`baltimore`.`district`
);
UPDATE `baltimore`
   SET `baltimore`.`neighbour` = (
   SELECT `crime_neighbourhood`.id
   FROM `crime_neighbourhood`
   WHERE
`crime_neighbourhood`.`neighbourhood` =
`baltimore`.`neighbour`
);
UPDATE `baltimore`
   SET `baltimore`.`crime_time` = (
   SELECT `crime_time`.`crime_time`
   FROM `crime_time`
   WHERE `crime_time`.`rec_id` =
`baltimore`.`id`
);
UPDATE `baltimore`
   SET `baltimore`.`crime_day` = (
   SELECT `crime_date`.`crime_date`
   FROM `crime_date`
   WHERE `crime_date`.`rec_id` =
`baltimore`.`id`
);

SELECT `crime_district`.id,
`baltimore`.`district` AS bdistrict,
`crime_district`.`district` AS cdistrict
   FROM `crime_district`,`baltimore`
   WHERE `crime_district`.`district` =
`baltimore`.`district`;
```

Figure 5: Structural Query Statements that Transformed Categorical Attributes to Numerics

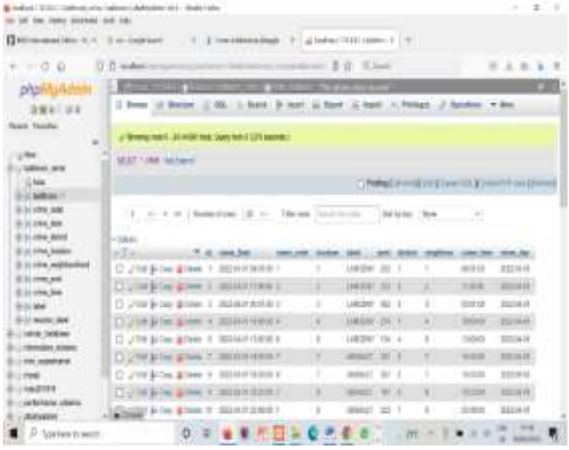


Figure 6:; The Indexed Mapped Ordinal Baltimore_Crime Database

3.3. Data Tansformation and Model Building

The data science programming language used was Python, wih a scikit library known as ordinal encoding class, to perform standard features encoding. The missing values were treated with SimpleImputer with a systematic strategy, a library in the same package. The models were built using the two separated formatted datasets. The Decision Tree (DT), Random Forest (RF), K-Nearest Neighbour (KNN) and Support Vector Machine (SVM) were trained to test how efficient the new data transformation technique would fare.

The classification concept in machine learning is about building models that separate data into distinct classes. The algorithms read 80% of the CSV data pre-labelled as the training set. This withheld 20% set was used to check the accuracy of the built classifiers to detect known intrusion instances. The models predicted the class membership of the instances based on what they learned from the training set. The classification outcomes of the models constructed were checked against their effectiveness in mining criminal patterns of attacks in each location. In short, the performance of these models was examined on the following standard parameters: accuracy, precision, recall and the f-score on the two main encoded dataset methods, Map Ordinal Encoding Method (MOEM) and Index Mapped Ordinal Encoding Method (IMOEM). The target are: LARCENY, ASSAULT and ROBBERY. The entire process is presented in Figure 7.

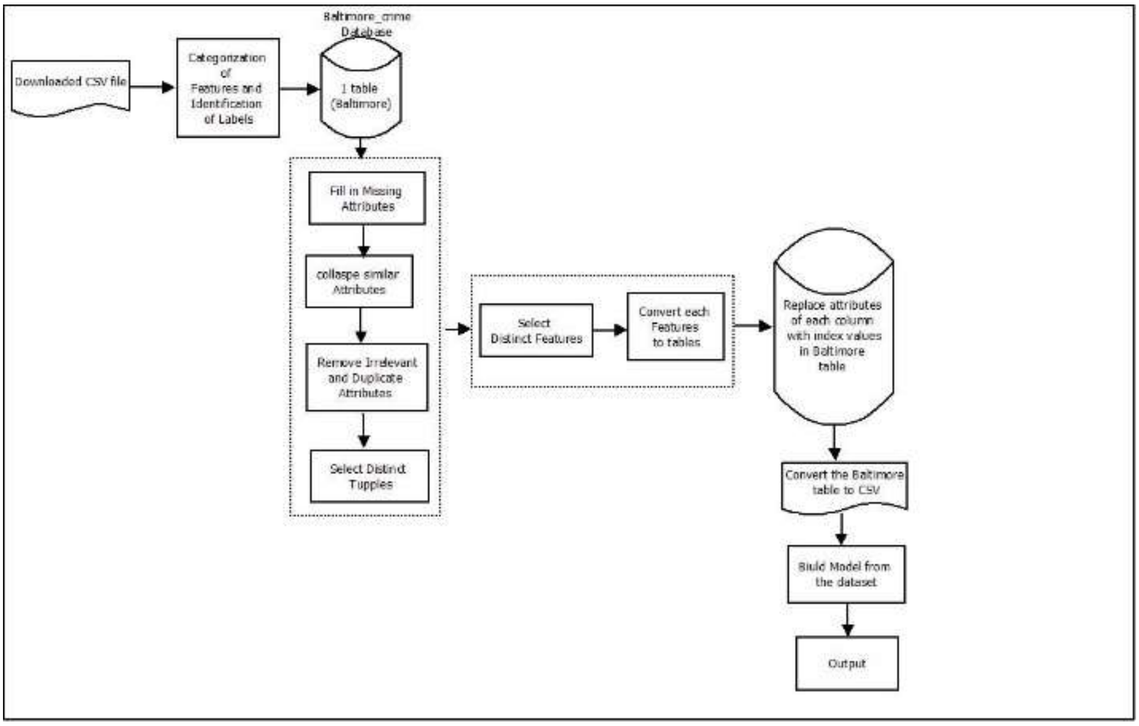

Baltimore_crime Database
Downloaded CSV file
Categorization of Features and Identification of Labels
1 table (Baltimore)
Fill in Missing Attributes
collaspe similar Attributes
Remove Irrelevant and Duplicate Attributes
Select Distinct Tupples
Select Distinct Features
Convert each Features to tables
Replace attributes of each column with index values in Baltimore table
Convert the Baltimore table to CSV
Biuld Model from the dataset
Output

Figure 7: The Index Mapped Ordinal Encoding Method (IMOEM) for Data Transformation Model

3.4 Methods of Dataset Splitting in Decision Tree Construction

The decision trees are hierarchical in structure. It is a tree-like pattern that involves partitioning of a dataset into different groups. Some statistical theories determine the partition feature to construct the tree from the root to the nodes down to the leaves. However, there are varieties of methods for selecting how to partition the transformed data. In a nutshell, decision trees require a robust mechanism to divide data sets into sections resulting in an inverted decision tree with root nodes at the top. Through the pass-over nodes of the trees, the layered model of the decision tree leads to the end outcome. Among these techniques are Entropy/Information Gain and Gini Index. The entropy as Eqn. (i) is a measure of purity or the degree of a random variable's uncertainty, impurity, or disorder. It is, in essence, the assessment of impurity or unpredictability in data points. However, when all elements in the data belong to the same class, the distribution is referred to as "Pure". This is essential in calculating Information Gain when choosing the best splitting feature, as found in Eqn. (ii).

.Entropy = $\sum_{i=1}^{c} P_i * \log_2 P_i$ equation (i)

Information Gain = 1 – Entropy equation (ii)

Given a probability distribution such that $P_i = \{P_1, P_2, P_3, \ldots\ldots\ldots\ldots\ldots\ldots P_n,\}$

Where (P_i) is the probability of a data point in the subset of D_i of dataset D.

Alternatively, Gini Index, as in Eqn. (iii), was also used in place of entropy to determine the level of information available in each feature in order to select the best splitting point. Gini index and entropy are metrics used to measure the level of information provided by an item of data when constructing a decision tree. The decision tree then leaves the split that results in the highest information gain as the best choice.

Gini = 1 - $\sum_{i=1}^{n} (P_i)^2$ equation (iii)

3.5. Model Evaluation Metrics

The model evaluation metric is a statistical quality measurement of how a model is fitted to the transformed dataset to predict the outcome of events. The evaluation metrics in different statistical concepts were used to test the performance of the models. The metric evaluation method for classification algorithms include accuracy, correlation matrix and confusion matrix while classification reports are precision, recall and accuracy score [12].

4.0 Results and Discussion

4.1 Confusion Matrix Reports

The effectiveness of one ordinal encoder and indexed identifiers encoding techniques were tested on the transformed Baltimore's dataset. The models; Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbour (KNN) and Decision Tree (DT), trained on the data derived from the two transformational methods and patterns, were extracted. As indicated in Table 5, the models used one ordinal encoding transformed dataset to classify the dataset. The results produced three distinct classes. The outcomes of the classification generated the following reports: precision, recall, f1-Score and the accuracy for each of the classifiers. The precision values for the four classifiers were perfect except for the KNN which had 10 % false alarm. The same KNN also had 89% values for both the recall, f1-Score as well as the overall accurate prediction of the test set.

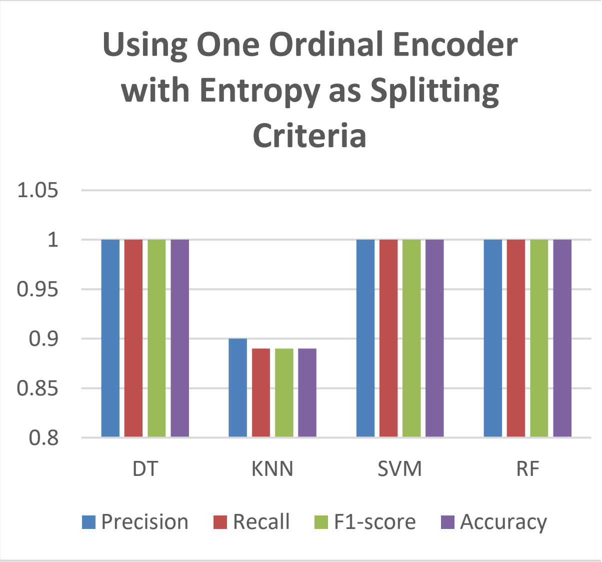


Figure 8: Map Ordinal Encoder Method (MOEM) Dataset Performance Using Entropy Splitting Indices

The performance of the MOEM data was good for building models as presented in Fig. 8. The entropy splitting criteria were used initially to perform the splitting process. Thus, the results for these three models (DT, SVM and RF) were 100% in precision, recall, f1-score and accuracy values respectively while KNN had 0.89 (89%) with 11% false alarm.

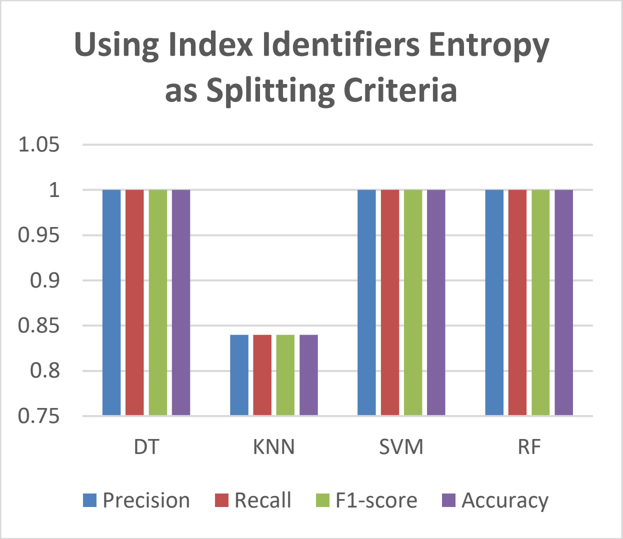


Figure 9: Index Mapped Ordinal Encoding Method (IMOEM) Dataset Performance Using Entropy Splitting Indices

The dataset encoded by the IMOEM was used in Figure 9 to train the same set of models as in Figure 8. In a similar way, entropy splitting indices were chosen to select the best partition point. Consequently, the experiments produced very similar results. The models' performance as in DT, SVM and RF models was perfect having 100% precision, recall, f1-score and accuracy results. However, the dataset did not fit well with KNN having 0.84 (84%) with 16% false alarm.

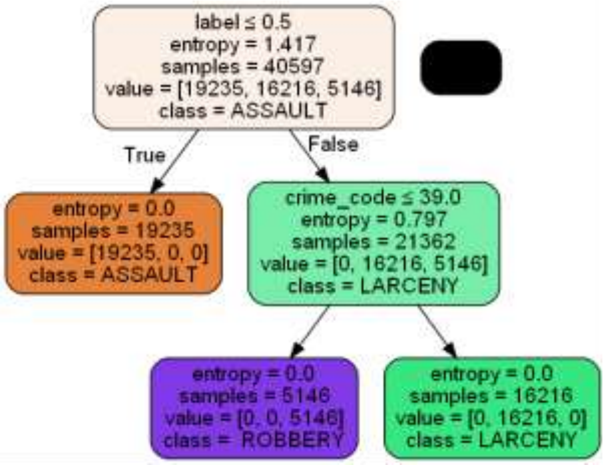


Figure 10: Decision Tree of Baltimore Dataset Using Entropy for Impurity Check

The decision tree in Fig. 10 vividly shows that the feature, “Assault”, has the highest entropy value of 1.417 when compared with the other features in the dataset. Hence it was selected as the root of the tree. The entropy values of the next splitting point showed that the total samples of 19,235 in the group were all under the class label, ASSAULT, which is often referred to as a pure classification. In the same level, 21, 362 of the datasets in this group had 0 number been classified as “Assault”; “Larceny” was 16, 218 while 5,146 were also classed as “Robbery”. However, three leaves were finally generated: “Assault”, “Robbery” and “Larseny”. The labels, “Assault” and “Larceny” had 0 member data classified under them while all the data were classified as “Robbery”. The third leaf also had all the instances in the group as Larceny with others having zero classification.

Gini Index

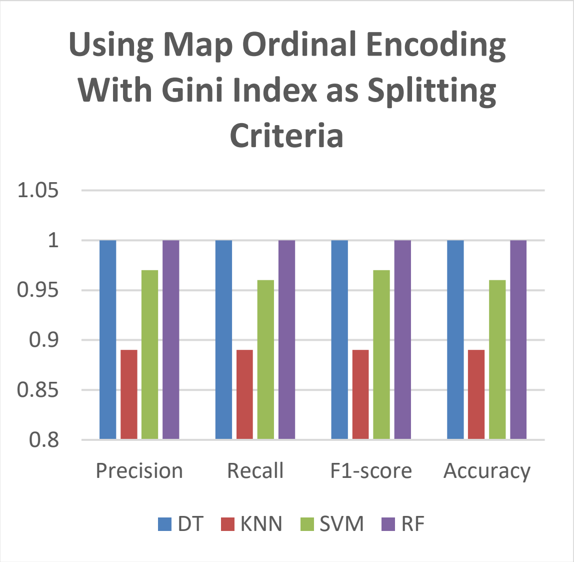


Figure 11: Map Ordinal Encoder Performance Using Gini Index Splitting Indices

In an attempt to probe further the quality of the split, another approach was implemented. The Gini Index is one of the reliable splitting indices. The Gini Index concept was introduced to determine the volume of impurity in the MOEM dataset; it was an effort to split at the best feature. The performance in Fig. 11 clearly revealed that the Decision Tree (DT) and Random Forest (RF) that were perfectly classified under entropy experiments also performed in a similar way. The two machine learning models perform extremely well with no false alarm recorded. However, the other models, Support Vector Machine(SVM) and K Nearsest Neighbour (KNN), also had relatively fair results of 96% and 89%, respectively.

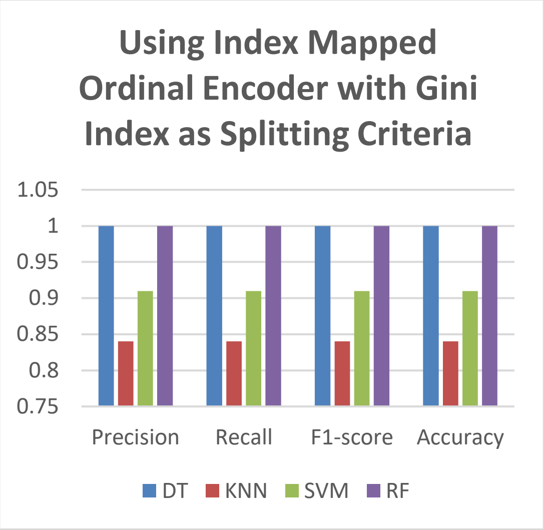


Figure 12: Index Mapped Ordinal Encoder Performance Using Gini Index Splitting Indices

The dataset encoded by IMOEM was used to build models using the Gini Index Impurity Measurement Criteria. The results of the models trained are as depicted in Fig. 12. These outcomes indicate that DT and RF performed perfectly without any false call while SVM and KNN performed fairly well in prediction with 91% and 84% respectively.

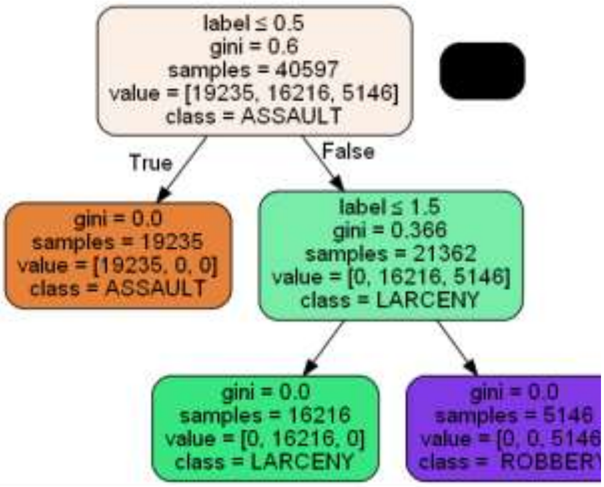


Figure 13: Decision Tree of Baltimore Dataset Using Gini Index for Impurity Check

Information Gain

The decision tree in Fig. 13 depicted "Assault" as the best quality feature with 0.6 Gini Index Value when compared to others at the start of constructing the model. Therefore, "Assault" was selected as the root of the tree. The next level of the tree generated two nodes, "Assault" and "Larceny". Howbeit, 19, 235 instances in this group were all under the class label "Assault", which simply indicates a pure result. The value of the Gini Index remained 0 and the node became a leaf. Since there was no alternative. The other node in this level had 0.366 Gini Index Value. This shows that it was not a pure partition. Out of the total samples of 21,362 in this group, none belonged to "Assault"; 16, 21 belonged to were "Larceny" while 5,146 were in the category of "Robbery". Therefore, the selected class for this node was "Larceny" based on the volume of the tupples categorised under it. The subsequent split produced two leaves. The first leaf, "Larceny", had 0 Gini Value andtupples of 16, 216 while the second leaf, "Robbery", has a total samples of 5,146.

4.7 Discussion of Results

Good quality data become imperative and a basic building block of any Machine Learning (ML) pipeline. The rating of these ML models can be affected by the state of their training data [18]. The use of inappropriate data preparation processes would significantly distort the results of predictions [18]. Thus, the output of this research work validate this assumption. The efficiency of the two dataset encoding techniques, MOEM and IMOEM, provide an improved alternative process. The data processing techniques trained the new models; decision tree, support vector machine and random forest and the ouput was 100% precision, recall, f-score and accuracy results. This is evident of an exemplary method. According to Stevens [28], Artificial Intelligent (AI) model's explainability depends on the data quality. If the data is not qualitatively rich, it will result in inaccurate insights that lead to wrong decisions in the human context.

According to Bertossi [8], both the model and post-hoc approaches of explainability depend on data input, feature importance, predictions and business rules. Explainability depends on the extent to which the data is qualitative, quantitative or biased for the AI model. In spite of the different impurity measurement parameters, the models in Figs. 8 and 9 generated a consistent performance, especially with decision tree and random forest models. Moreover, they are models known for clarity, ease of interpretation and being simple to follow by human beings. The qualities of these classifiers make the experiment worthwhile, as presented in the diagrammatic results (Figs. 7 and 13).

However, the techniques did not perform well with the K-Nearest Neighbour. The IMOEM application, though a relational database, and Mysql, the topmost mining database, reported by Munoz-Gama [21], conveniently encoded raw data directly in the database for model building. The digital nature of the dataset used to train the models improved the memory management usage of devices and the speed of predictions.

Considering the submission of Stevens [26], RFs are the most consistent ensemble classifiers and this feature makes them to be ranked among the best. The performance of Random Forest as an ensemble model has always been precised and accurate. The outcome of this research work further buttresses the position of this claim. These models performed exceptionally well during classification processes as seen in Figs.xi and xii.

Similarly, the performance of the Decision Tree Models (DTMs) built using the IMOEM encoding technique also supports the assertion of Sarker [25] that ML detection systems on cyber threats, such as DTMs are effective for prediction. It indicates that DTMs are ranked higher than other non-DT-based algorithms when it comes to detecting new intrusion patterns. According to Sarker [25], the importance of feature selection in model building, especially DTMs, cannot be overstated. As a tree-based intrusion detection model, new models are trained based on the significance of the features. In light of this, the two feature selection techniques (Entropy and Gini index) resulted in similar performance as shown in Figs. viii, ix, xii, and xiii to reinforce the claim made by the authors. The outcomes of the predictions regarding precision, recall and accuracy show this.

In the same vein, the split's quality is paramount to the performance of the models. The Gini Index and Entropy were used to determine the level of impurity and information available in each feature. [9] and [28] state that a good feature selection process would result in efficient machine learning models. Moreso, emphasis has been placed on the representation of the real-world events in a unambiguous way. However, this can only be achieved using appropriate pre-processing machine tools when constructing the models for prediction as reported by Patel [16].

Summarily, the results of this research confirm the findings of Devi and Suganthe [12] which state that Vector Machines (SVM) and Random Forest (RF) are among the most used methods due to its high accuracy detection and efficient memory usage. This analysis also provides insight into AI and data science as a roadmap to detect threats based on attacks in a community.

4.3. Limitation of these Results

The users of Index Map Ordinal Encoding Method (IMOEM) are expected to have a relative knowledge of Structural Query Language (SQL).

5.0 Conclusion

The dataset encoded by both IMOEM and OOE to train the following models, DT, SVM, KNN and RF, generated the same outcomes. The results were perfect for DT, SVM and RF in terms of precision, recall, f1-score and accuracy values This simply established that IMOEM is a good data transformation model. One of the core observations were that the performance was perfectly similar to the decision tree based models.

Comparatively, IMOEM repeated the same output with MOEM when Gini Index splitting criteria were used. The two methods produced another perfect results with respect to precision, recall, f1-score and accuracy values without any false alarm. This is a demonstration of consistency in performance and this equally enhances the reputation of IMOEM as an attractive model for data encoding during the pre-processing phase in terms of prediction. Subsequently, the IMOEM and MOEM's performance reaffirms the fact that Decision

Tree based models are attractive and perfect for prediction and classification tasks.

Finally, the people of Baltimore City in the United States of America faced more assaults than larceny or robbery. Hence, the city's crime control officers should place a premium on assaults of citizens of the city.

5.1 Recommendation

It would be a good idea to keep the dataset in the original storage (database) or at the edge when carrying out data transformation to avoid damaging or loss of data items during the pe-processing stage.

References

1. Abdullahi, M., Baashar, Y., Alhussian, H., Alwadain, A., Aziz, N., Capretz, L. F., & Abdulkadir, S. J. (2022). Detecting cybersecurity attacks in the Internet of Things using Artificial Intelligence methods: A systematic literature review. *Electronics*, 11(2): 198, 1-27.
2. Alasadi, S. A., & Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences*, 12(16), 4102-4107.
3. Alloghani, M., Al-Jumeily, D., Hussain, A., Mustafina, J., Baker, T., & Aljaaf, A. J. (2019). Implementation of Machine Learning and Data Mining to Improve Cybersecurity and Limit Vulnerabilities. *Nature-Inspired Computation in Data Mining and Machine Learning*, 855, 47-76.
4. Alshaibi, A., Al-Ani, M., Al-Azzawi, A., Konev, A., & Shelupanov, A. (2022). The comparison of cybersecurity datasets. *Data*, 7(2), 1-18.
5. Alsoufi, M. A., Razak, S., Siraj, M. M., Nafea, I., Ghaleb, F. A., Saeed, F., & Nasser, M. (2021). Anomaly-based intrusion detection systems in iot using deep learning: A systematic literature review. *Applied Sciences*, 11(18): 8383, 659-675.
6. Anindya, Mozumdar (2020). A guide to encoding categorical features using R. https://www.r-bloggers.com/2020/02/a-guide-to-encoding-categorical-features-using-r/accessed 29 April, 2022.
7. Ayinla, I. B. and Akinola, S. O. (2020). An improved collaborative pruning using ant colony optimization and pessimistic technique of C5.0 decision tree algorithm, *International Journal of Computer Science and Information Security*, 18(12), 111-123.
8. Bertossi, L., & Geerts, F. (2020). Data quality and explainable AI. *Journal of Data and Information Quality (JDIQ)*. 12(2), 1-9.
9. Cabitza F, Campagner A, & Ciucci D. (2019). New frontiers in explainable AI: Understanding the GI to interpret the GO LNCS, vol. 11713. In: *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Cham: Springer; (p. 27-47).
10. Cremer, F., Sheehan, B., Fortmann, M., Kia, A. N., Mullins, M., Murphy, F., & Materne, S. (2022). Cyber risk and cybersecurity: A systematic review of data availability. *The Geneva Papers on Risk and Insurance-Issues and Practice*, 47, 698–736.
11. Damian Chan (2020) Data Transformation for Machine Learning, https://insidebigdata.com/2020/05/07/data-transformation-for-machine-learning/ accessed 30 April, 2022.
12. Devi, E. M., & Suganthe, R. C. (2017). Feature selection in intrusion detection grey wolf optimiser. *Asian Journal of Research in Social Sciences and Humanities*, 7(3), 671-682.
13. Dilek, S., Cakır, H., & Aydın, M. (2015). Applications of Artificial Intelligence techniques to combating cyber crimes: A review. *International Journal of Artificial Intelligence & Application*, 6, 21–39.
14. Frank, E., & Hall, M. (2001). *A simple approach to ordinal classification*. In European conference on machine learning (pp. 145-156). Springer, Berlin, Heidelberg.
15. Han, J., Kamber, M., & Pei, J. (2011). *Data mining: Concepts and. techniques* (3rd ed), Morgan Kauffman.
16. Harshil Patel (2021), What is Feature Engineering — Importance, Tools and Techniques for Machine Learning, https://towardsdatascience.com/what-is-feature-engineering-importance-tools-and-techniques-for-machine-learning-2080b0269f10, accessed 29 April, 2022.
17. Jason, Brownlee (2020), *Ordinal and One-Hot encodings for categorical data*. https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/accessed 02 May, 2022.
18. Grus, J. (2019). Data science from scratch: First principles with python (2nd ed.). O' Relly.
19. Li, C. (2019). *Preprocessing methods and pipelines of data mining: An overview*. arXiv preprint arXiv:1906.08510.
20. Mayur, B. (2021). *How to perform One-Hot encoding for multi-categorical variables*, https://www.analyticsvidhya.com/blog/2021/05/how-to-perform-one-hot-encoding-for-multi-categorical-variables/ accessed 29 April, 2022.

21. Munoz-Gama, J., & Lu, X. (2022). *Process mining workshops*: ICPM 2021 International Workshops, Eindhoven, The Netherlands, October 31–November 4, 2021, Revised Selected Papers.
22. Naik, B., Mehta, A., Yagnik, H., & Shah, M. (2021). The impacts of artificial intelligence techniques in augmentation of cybersecurity: a comprehensive review. *Complex & Intelligent Systems*, 1-18.
23. Nwaiwu, J. C., & Imafidon, H. C. (2017). Knowledge management and organisational survival: a study of telecommunication industry in Port Harcourt, Nigeria. *International Journal of Advanced Academic Research*, 3(7), 40-53.
24. Peter Blumberg(2020), https://www.bloomberg.com/news/articles/2020-02-08/facebook-vows-to-improve-security-after-hack-of-29-million-users accessed 10 April, 2022.
25. Sarker, I. H., Abushark, Y. B., Alsolami, F., & Khan, A. I. (2020). Intrudtree: a machine learning based cyber security intrusion detection model. *Symmetry*, 12(5), 754.
26. Seveso, A., Campagner, A., Ciucci, D., & Cabitza, F. (2020). Ordinal labels in machine learning: a user-centered approach to improve data validity in medical settings. *BMC Medical Informatics and Decision Making*, 20(5), 1-14.
27. Sikibi, M. (2022). Use Data Mining Cleansing to Prepare Data for Strategic Decisions. In Data Mining-Concepts and Applications. *IntechOpen*.
28. Stevens, A., De Smedt, J., & Peeperkorn, J. (2022). *Quantifying Explainability in Outcome-Oriented Predictive Process Monitoring*. In International Conference on Process Mining (pp. 194-206). Springer, Cham.
29. Sunil R. (2015). *Simple Methods to deal with Categorical Variables in Predictive Modeling*, https://www.analyticsvidhya.com/blog/2015/11/easy-methods-deal-categorical-variables-predictive-modeling/accessed 01 May, 2022.
30. Tung. M. Phung (2019). *How to convert Categorical Variables to Numerical Variables*, https://tungmphung.com/how-to-convert-categorical-variables-to-numerical-variables/ accessed 29 April, 2022.
31. Zuar Team (2022), https://www.zuar.com/blog/data-transformation-types-process-benefits-and-definition/ accessed 30 April, 2022.