information

MDPI

Article

A Machine Learning-Based Method for Content Verification in the E-Commerce Domain

Theodoros Alexakis, Nikolaos Peppes, Konstantinos Demestichas and Evgenia Adamopoulou *

Institute of Communication and Computer Systems, National Technical University of Athens,
15773 Athens, Greece; talexakis@cn.ntua.gr (T.A.); npeppes@cn.ntua.gr (N.P.); cdemest@cn.ntua.gr (K.D.)
* Correspondence: eadam@cn.ntua.gr; Tel.: +30-210-772-1478

Abstract: Analysis of extreme-scale data is an emerging research topic; the explosion in available data raises the need for suitable content verification methods and tools to decrease the analysis and processing time of various applications. Personal data, for example, are a very valuable source of information for several purposes of analysis, such as marketing, billing and forensics. However, the extraction of such data (referred to as person instances in this study) is often faced with duplicate or similar entries about persons that are not easily detectable by the end users. In this light, the authors of this study present a machine learning- and deep learning-based approach in order to mitigate the problem of duplicate person instances. The main concept of this approach is to gather different types of information referring to persons, compare different person instances and predict whether they are similar or not. Using the Jaro algorithm for person attribute similarity calculation and by cross-examining the information available for person instances, recommendations can be provided to users regarding the similarity or not between two person instances. The degree of importance of each attribute was also examined, in order to gain a better insight with respect to the declared features that play a more important role.

Keywords: machine learning (ML); deep learning (DL); person similarity; content verification; feature importance; person fusion

check for updates

Citation: Alexakis, T.; Peppes, N.; Demestichas, K.; Adamopoulou, E. A Machine Learning-Based Method for Content Verification in the E-Commerce Domain. *Information* **2022**, *13*, 116. https://doi.org/10.3390/info13030116

Academic Editor: Gabriele Gianini

Received: 14 December 2021
Accepted: 21 February 2022
Published: 26 February 2022

1. Introduction

1.1. Overview

The ongoing Fourth Industrial Revolution has shifted everyday human activities to a more digitized nature. Societies across the world are becoming increasingly digitized in a wide spectrum of their activities, e.g., financial transactions, communication, social interactions and work. The continuous digitalization in our hyper-connected society has enabled the generation of vast volumes of data. The global internet traffic has increased dramatically over the last 30 years and still continues its uprising trend. According to CISCO [1], the annual network traffic for 2020 was 2.3 zettabytes or 61,386GB per second. The projection of the global internet traffic according to the World Data Bank is that, in 2022, it will reach 150,000 GB per second [2].

This explosion of data generation in recent years has led to the emergence and establishment of big data technologies which tend to substitute the former dominant data management systems such as the typical relational databases (Structured Query Language (SQL)). The exploitation of big data technologies is not without its challenges for many IT solution providers across various domains. Among the most significant issues that big data adopters must overcome are the structure of data, the semantic information hidden in unstructured data, the mining of knowledge residing in them, etc. [3]. Quite often, these vast amounts of data include data that refer to persons. Due to the many different attributes that refer to the same person, it is very common for organizations and data controllers to keep duplicate instances that, in some cases, may be identical but, in most, differ slightly and could, thus, be mistakenly treated as referring to different persons. Furthermore, as data volumes grow, storage also needs to increase, rendering the minimization of storage space a key challenge in order to build more efficient backup processes [4].

In this light, efficient handling of vast amounts of data is crucial, since duplicate person instances, for example, can cause problems in machine learning (ML) and deep learning (DL) classifiers as they can have an impact on the training speed or the quality of learned models, thus affecting the efficiency of data analysis. Therefore, the detection and elimination of duplicate instances in complex big data settings becomes a necessity, especially as the growth of data volumes continues at a rapid pace. The merging of duplicate instances, often called data fusion, is the combination of data from heterogeneous sources in order to achieve an improved accuracy compared to the use of a single source alone [5].

Data fusion is a traditional method for processing massive sets of data flows in e-commerce systems. Increased total costs and high energy consumption are two main drawbacks of this conventional method. Data fusion processes can be part of a content verification method in big data flows. Performance metrics of these processes can subsequently be combined in order to provide accurate feedback for supporting decision making. More specifically, e-commerce systems involve heterogeneous data sources, ranging from physical sensors (such as trackers of environmental conditions or vehicle GPS) to digital sources (multimedia content, textual data, financial reports, etc.). This approach can enable energy savings and a reduction in costs during the integration phase, in e-commerce systems. Figure 1 depicts the process flow of the described methodology for data fusion in an e-commerce system.

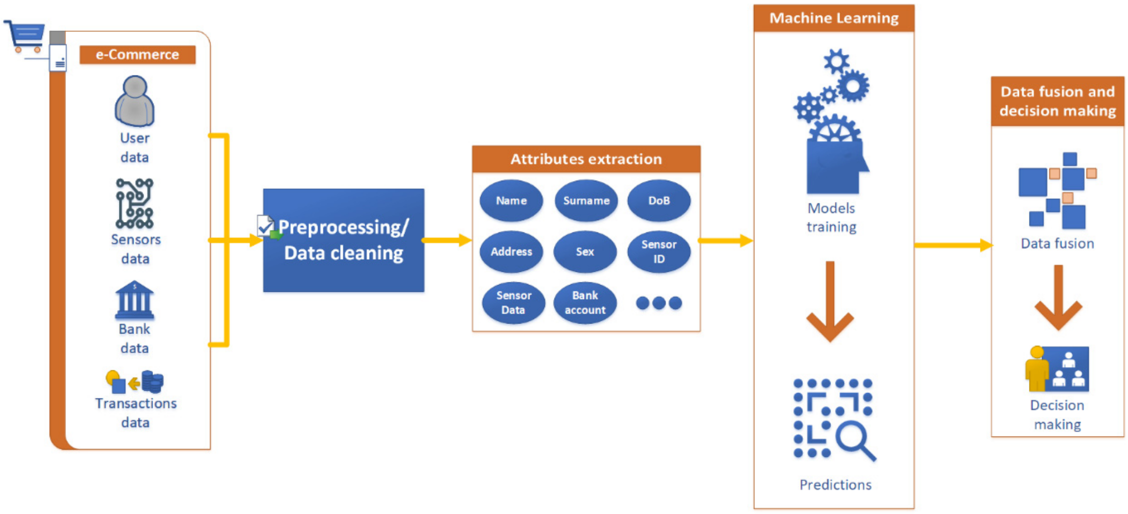

Figure 1. Process flow for data fusion in an e-commerce system.

This study aimed to explore the usage of pretrained machine learning and deep learning models to detect similar instances of persons, consisting of five distinct features per record. In order to achieve this, a synthetic dataset of random person instances was created and, using the Jaro similarity algorithm, each person instance was compared to every other person instance. Through this similarity calculation process, a new training dataset was produced to be provided as input to ML and DL algorithms. Additionally, the importance of each was examined so as to gain better insight into how each feature affects the duplication detection process.

The remainder of Section 1 features related research. Data exploration together with the description of the proposed methodology is carried out in Section 2. Section 3 presents the analysis and the results acquired, while Section 4 reasons on them, and Section 5 concludes the study.

1.2. Related Works

The e-commerce domain is rapidly evolving and expanding as it engages an increasing number of state-of-the-art technologies for efficiently handling the vast amounts of heterogeneous data collected through various sources [6]. These state-of-the-art technologies are mainly based on big data analytics, as well as machine learning (ML) and deep learning (DL) algorithms. In this direction, there are several academic and research efforts which apply ML and DL technologies in the supply chain and e-commerce domain.

The supply chain contains a wide variety of processes, starting from the production of goods until their delivery to end customers. ML can be applied throughout the entire supply chain. Various studies offer ML-based solutions to different problems, such as for the reliable monitoring of motor status [7] and for transportation and distribution monitoring by enabling automated routing of autonomous or light vehicles [8]. E-commerce, as part of the supply chain management, includes different aspects and functionalities where ML models can provide benefits. In this direction, an extensive literature review concerning the e-commerce initiatives that can benefit from ML was carried out by Policarpo et al. [9]. According to this study, there are eight main e-commerce goals that ML models can provide benefits to, namely: (i) recommendation systems; (ii) fraud detection; (iii) customer relationship management; (iv) marketing campaigns; (v) repurchase prediction; (vi) discovering relationships between data; (vii) purchase prediction; and (viii) product return prediction. As is obvious, most of the aforementioned goals are directly connected with customer behavior. Furthermore, the same study [9] featured an analysis of the most popular ML algorithms used in e-commerce, including random forests, support vector machines (SVMs) and neural networks (NNs).

In order to monitor and assess customer behavior, personal attributes such as name, surname and address are of high importance. Following the rapid growth of e-commerce, relevant datasets are vast, making it important to avoid duplicate records. In this light, there are several research efforts on record de-duplication using ML or DL models. Carvalho et al. [10] explored the de-duplication process by engaging ML and, specifically, genetic programming. Their approach aimed to unburden users from choosing an appropriate de-duplication function by extracting knowledge for their configuration from the data fed to their model [10]. Moreover, Christen and Goiser [11] introduced an ML-based method using a decision tree classifier for twelve different artificially generated datasets. The results of this study were quite promising since, for all the experiments conducted, the accuracy metric scored above 90%. Additionally, Elfeky et al. [12] introduced the TAILOR framework which serves for the record linkage or data de-duplication process. Their experimental results showed that ML methods outperformed probabilistic methods [12]. Despite the promising results of [12], the authors acknowledged the difficulty in obtaining annotated datasets. Based on this, Gschwind et al. introduced their proposed solution which comprises rule-based linkage algorithms and ML models. Their study achieved a 91% recall rate on a real-world dataset [13].

The efficiency of ML classifiers is often highly dependent on the features of the data given as input [14]. Thus, feature importance and feature selection are gaining increasing ground for ML and DL solutions, in order to increase the quality of the produced results, and to reduce computational costs [15]. The calculation of feature importance can enable effective feature selection, improving the computational performance of ML models. For example, such a feature selection process was used to improve the computational efficiency of chatter vibration diagnosis by Tran et al. [16]. Another method engaging feature importance calculation is 'feature weighting'. A comprehensive review about feature weighting techniques and their characteristics was performed by Iratxe et al. [17].

2. Materials and Methods

The main goal of this study was to propose an efficient solution to detect duplicate person instances, which can be useful in e-commerce applications, as discussed above. These instances may include various types of distinctive information such as first name, surname, date of birth (DoB), residence and sex. This study continued the work performed in [18], where the comparison of person instances was conducted by comparing only the first name and surname of each pair of persons using the Jaro [19], Jaro–Winkler [20], Levenshtein [21], cosine similarity [22] and Jaccard similarity [23] techniques. Thus, this study is a step forward, with respect to [18], as it engages more attributes for comparison purposes as well as ML and DL model performance evaluation, so as to enable the creation of a near-real-time application for duplication detection.

2.1. Dataset Exploration

The dataset used for the purposes of this study contains 100 person instances which have been synthetically and randomly created. Each person instance consists of five different attributes and is compared to every other person instance so as to calculate the similarity degree. The five distinct characteristics are: (i) first name; (ii) surname; (iii) date of birth; (iv) address; and (v) sex.

The dataset provided as input contains randomly generated person instances and approximately 20% of similar person instances. Every person instance in the dataset can be compared to every other person in the dataset, and, thus, a total number of 4950 similarity calculations were performed. The number of calculations can be computed by Equation (1), which practically calculates the number of possible combinations [24].

$$\mathrm{C(n,r)} = \begin{pmatrix} \mathrm{n} \\ \mathrm{r} \end{pmatrix} = \frac{\mathrm{n!}}{(\mathrm{r!(n-r)!})} \tag{1}$$

where:

- C is the total number of calculations;
- n is the number of instances contained in the dataset;
- r is the number of instances compared in each calculation.

Thus, since we have 100 instances (n = 100) and every comparison includes 2 persons (r = 2), Equation (1) leads to the identified 4950 comparisons.

The 4950 comparisons take into account all five attributes of each person instance and treat them as one single record. The choice to utilize a synthetically created dataset of random person instances, including a predefined percentage of similar records, was intentionally made so as to offer a better insight into the functionality of the algorithms and methods used for this study. In this light, the following preprocessing steps were executed for the 100 random persons dataset so that it could be further used as input to the Jaro algorithm in order to perform the aforementioned 4950 comparisons:

- Name normalization: One of the most common and difficult issues while dealing with consolidating data from multivariate and heterogeneous sources is the existence of different features in the consumed data instances. Text normalization deals with the processes of transforming the original raw text into a canonical form, which is different from the initial one. Multiple methods can be utilized to transform the raw data, including Unicode quirks, upper to lower case conversion, irrelevant symbol removal, whitespace collapsing and normalization and conversion of each distinct word to a double metaphone. Additionally, the detachment of special characters (umlauts, rare letters, accents and/or other non-typical Unicode-based normalizations) and stop words (e.g., string punctuations) is part of this step, using the appropriate libraries.
- Greed matching of the pairs: the next step involves ignoring multiple matches, the detachment of duplicates and eventually the concatenation of the remaining pairs of persons.

The calculation of the Jaro similarity metric can be performed by using Equation (2):

$$Jaro\ similarity = \begin{cases} 0,\ for\ m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right),\ for\ m! = 0 \end{cases} \tag{2}$$

where:

- m is the number of matching characters. The characters whose distance is not greater than the result of Equation (3) are considered as matching characters:

$$\left\lfloor \frac{\max(|s_1|, |s_2|)}{2} \right\rfloor - 1 \tag{3}$$

- t is half the number of transpositions. A transposition is considered to occur when two characters are the same but not in the same place in the two strings examined. Thus, the number of transpositions is half the number of the matched misplaced characters.
- $|s_1|$ and $|s_2|$ are the lengths of strings $|s_1|$ and $|s_2|$, respectively.

Moreover, three ML methods were considered for the purposes of this study: logistic regression (LR), a random forest (RF) and a penalized support vector classifier (SVC), together with a neural network (NN) as a DL method for the creation of a training dataset that can be divided into training and test data. For the creation of this training dataset, the Jaro technique featured very promising and computationally efficient results [18]. The previously discussed dataset of the 100 randomly created name instances was used as input to the Jaro algorithm, and a new dataset of 4950 instances was, subsequently, constructed. This dataset contains 4950 records (rows) and 6 columns: 1 for each of the 5 attributes containing the comparison result of this specific attribute between the two persons examined as well as a column with a label indicating whether this record contains the result of two similar persons or not (0 for non-similar and 1 for similar). This produced labeled dataset can be used as input to the classification ML algorithms presented in the next section. Table 1 features a snapshot of the generated dataset. As can be seen in Table 1, the attribute similarity results that belong to different persons (label = 0) are lower than the same results of similar persons (label = 1).

Table 1. Training dataset snapshot.

Name	Surname	Address	DoB	Sex	Label
0.0	0.0	0.53626	0.61851	0.63888	0
0.46666	0.48148	0.51024	0.53148	0.63888	0
1.0	0.96111	0.90648	0.97407	1.0	1

An issue worth mentioning considering the generated input training dataset is that it contains only ten records labeled as ‘1’, with the rest of them labeled as ‘0’. This practically reveals that the dataset is imbalanced, and, as a result, the performance of some ML algorithms will be affected. Further details about the algorithms’ behavior in such cases as well as the countermeasures taken to avoid this risk are presented, in detail, in the next sections.

2.2. Method Followed

The aim of the integration of machine learning techniques in the current study resides in the evaluation of a near-real-time algorithm in order to accelerate the processes of duplication detection and content verification of person instance data retrieved from multiple sources.

ML and DL techniques adopting training processes of appropriate models and generating ‘knowledge’ from data based on real-word experience can improve the overall performance and provide optimized solutions for content verification in large volumes of heterogeneous data. Even though the initial training stage can be a complex and time-consuming activity, once the model has been built, it can easily be applied to recognize or detect patterns of interest in newly provided data.

The first step in the process is to acquire the appropriate knowledge concerning the initial imported dataset with the 100 randomly created name instances. This dataset includes, as already mentioned, 10 pairs of similar name instances (persons), whereas the other 80 name instances are completely different to each other. Additionally, information such as the address, the date of birth (DoB) and the sex of the person instance is included.

The next step involves the application of the appropriate preprocessing methods, in order to split the first name and the surname from the initial concatenated feature into two distinct features, and to clean the raw dataset information using the name normalization process mentioned previously. Following the aforementioned processes, the resulting dataset consists of 100 random name instances, each of them including 5 (cleaned) different features: name, surname, address, date of birth and sex, as well as the label that separates the pair of similar person instances from the remaining ones.

Moving on, and in order to obtain the appropriate dataset used for the training of the ML models, the Jaro algorithm was applied following the results of [18]. Then, the five input features (name, surname, address, DoB and sex) were compared, and the training dataset mentioned in Section 2.1 with 4950 records was generated. The five input features receive values, as a result of the Jaro similarity comparison, in the range of zero to one.

As mentioned in Section 2.1, the generated dataset can be characterized as imbalanced. This means that a disproportionate rate of records is observed comparing the two classes inside the dataset; more specifically, of a total of 4950 records, 10 are labeled as similar (label equal to 1), whilst 4940 are labeled as different (label equal to 0). Imbalanced datasets represent a common issue in classification processes in ML techniques.

To handle the imbalanced class problem in the generated training dataset and to come up with the optimal solution regarding the ML model selection, the following techniques were implemented in the context of the present study:

1. Firstly, a conventional ML algorithm was selected, namely, the logistic regression algorithm, in order to train the model using the original, imbalanced dataset.
2. Consequently, an up-sampling method was applied to the original, imbalanced dataset. Up-sampling techniques are processes of randomly duplicating records from the minority label (class), in order to improve the model's extracted metrics and its overall performance in comparison with the original, imbalanced dataset [25–27]. The application of this type of method to the original dataset leads to the resampling of its initial records, setting the final number of the minority class samples (which correspond to duplicate entries) equal to the number of the majority class instances in the original, imbalanced dataset.
3. This generated, up-sampled dataset was applied again in the (initially selected) logistic regression algorithm in order to compare the extracted performance metrics with those extracted from the initial LR implementation in the original, imbalanced dataset.

Another suggestion for tackling the imbalanced class problem consists of down-sampling the majority class [28]. In principle, down-sampling selects arbitrary records from the majority label (class) and removes them from the original dataset, in order to resample it, without replacement processes. In a down-sampled dataset, the number of majority class records will be equal to the corresponding minority class of the original dataset. For this study, this type of approach was not followed, since the total number of records in the original dataset was already quite small; thus, if a down-sampling method was applied, the generated dataset would include twenty records in total, after arbitrary removal of instances from the majority label.

Another method that was implemented was the penalized learning algorithms technique [29]. These algorithms are applied to the original, imbalanced dataset and increase the cost of classification errors in the minority label. To penalize mistakes, we selected the appropriate arguments that enable probability estimation and a balanced class weight selection, in order to 'punish' more severe errors detected in the minority classes by a specific value corresponding, in measure, to how much these are under-represented in the overall dataset.

In terms of modern applied ML methods, tree-based classifiers consist of an ideal solution for imbalanced dataset classification [30,31]. Their hierarchical structure and the ability of cost incorporation in diverse types can yield a satisfactory performance on imbalanced datasets. In the current study, the random forest classifier that was selected and applied to the original, imbalanced dataset is an ensemble technique that usually outperforms the isolated decision tree-based algorithms.

The final implemented method includes the classification of the original and up-sampled datasets using deep neural network techniques in order to compare and assess the extracted performance metrics, before and after the resampling process, which was previously described. Figure 2 summarizes the procedures described in Section 2.2.

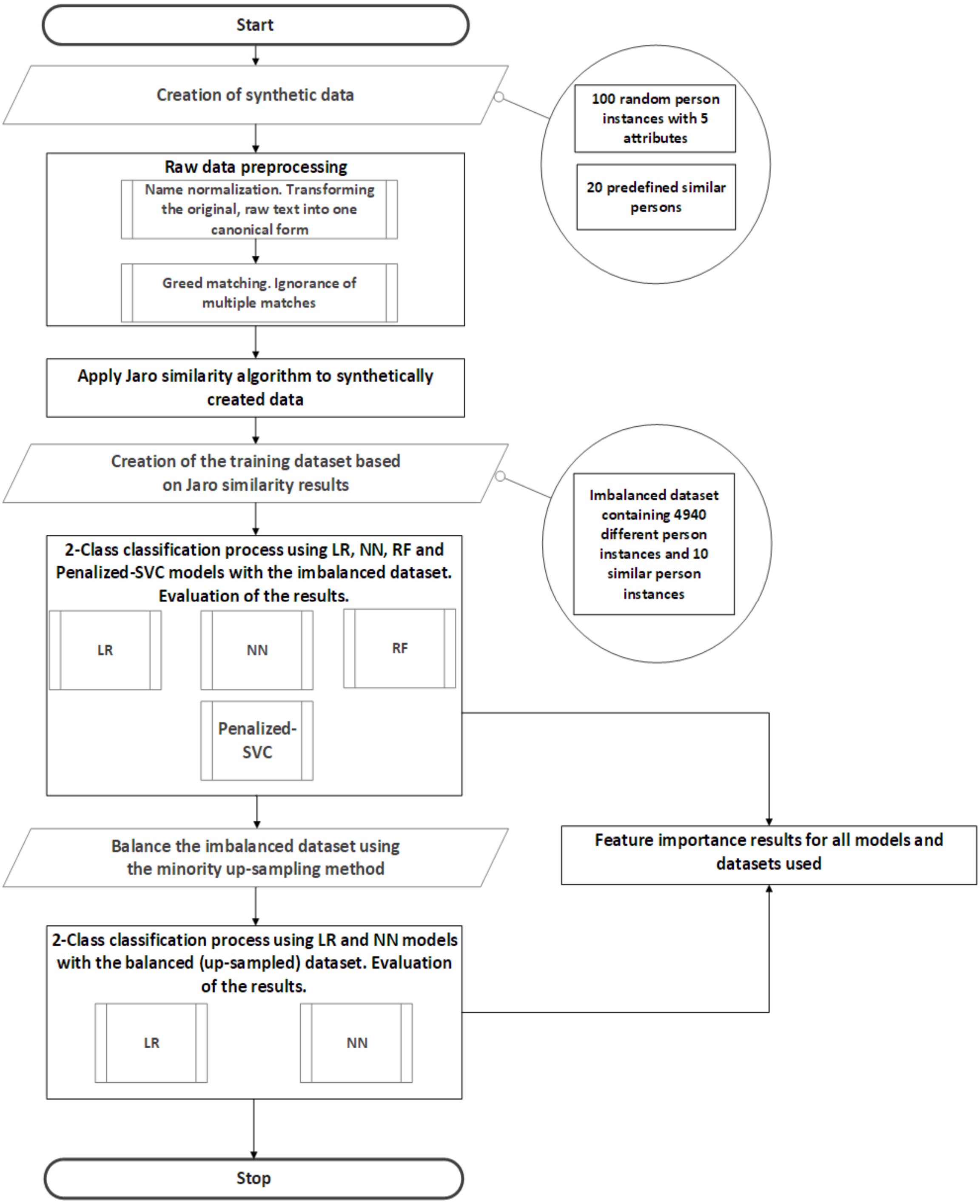
Start
Creation of synthetic data
100 random person instances with 5 attributes
20 predefined similar persons
Raw data preprocessing
Name normalization. Transforming the original, raw text into one canonical form
Greed matching. Ignorance of multiple matches
Apply Jaro similarity algorithm to synthetically created data
Creation of the training dataset based on Jaro similarity results
Imbalanced dataset containing 4940 different person instances and 10 similar person instances
2-Class classification process using LR, NN, RF and Penalized-SVC models with the imbalanced dataset. Evaluation of the results.
LR
NN
RF
Penalized-SVC
Balance the imbalanced dataset using the minority up-sampling method
Feature importance results for all models and datasets used
2-Class classification process using LR and NN models with the balanced (up-sampled) dataset. Evaluation of the results.
LR
NN
Stop

Figure 2. Methodology steps taken for this study.

Finally, in order to benchmark the performance of the proposed solutions and models developed for the needs of this study, Section 3 collects, compares and demonstrates the results in a detailed manner. In addition to the ML metrics, an analysis of the feature importance for each method used for both the imbalanced and up-sampled datasets is provided.

3. Results

In the previous section, a detailed description of both the data used and the methodology followed for the purposes of this study was provided. Following the aforementioned procedures, a series of results is produced and presented in more detail in this section. More specifically, this section is divided into three subsections: the first one presents the results for the ML algorithms that take the imbalanced dataset as input; the second one presents the results for the ML algorithms which use the up-sampled balanced dataset as input; and the last one presents the importance of the five features calculated using the coefficient calculation. The hyperparameters chosen for each of the examined models are presented in Table 2.

Table 2. ML and DL model parameters.

Classifier	Parameters
Logistic Regression (LR)	penalty = ‘elasticnet’, tol = 0.001, C = 1.0, class_weight = None, random_state = 1001, solver = ‘saga’, max_iter = 1000, verbose = 1001, n_jobs = -1, l1_ratio = 0.5
Random Forest (RF)	n_estimators = 500, criterion = ‘entropy’, max_features = ‘log2’, n_jobs = -1, random_state = 1002, verbose = 1
Penalized SVC	kernel = ‘linear’, class_weight = ‘balanced’, probability = True, penalty = ‘l1’, loss = ‘squared_hinge’, tol = 0.001, C = 2.0, multi_class = ‘ovr’, verbose = 1, random_state = 1002, max_iter = 10000
Neural Network (NN)	hidden layers = 3, model = ‘sequential’, input_dim = 5, activation_function = ’relu’, loss = ’binary_crossentropy’, optimizer = ‘adam’, metrics = [‘accuracy’]

The metrics used for the evaluation of the results were: (i) accuracy; (ii) loss; (iii) precision; (iv) recall; and (v) F1-score. The accuracy metric is defined as the ratio between the correct predictions over the total samples of the dataset and was calculated as shown in Equation (4) [32]:

$$\text{accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

where:

- *TP* stands for true positives, i.e., a record labeled as similar (label = 1) that, indeed, concerns two similar person instances;
- *TN* stands for true negatives, i.e., a record labeled as non-similar (label = 0) that, indeed, concerns two different persons;
- *FP* stands for false positives, i.e., a record labeled as similar (label = 1) that actually concerns two different persons;
- *FN* stands for false negatives, i.e., a record labeled as non-similar (label = 0) that actually concerns two similar person instances.

The loss metric is commonly used for the evaluation of ML and DL classification algorithms such as those engaged in this study. More specifically, the log loss (logistic loss or cross-entropy loss) metric was used, which represents the negative log likelihood of a logistic model that returns the predicted probabilities for its ground truth (correct) labels.

In the case examined, the labels are '0' and '1'; thus, the log loss was calculated as shown below (Equation (5) [33]:

$$\log\mathrm{loss}(\mathrm{y},\mathrm{p}) = -(\mathrm{y}\log(\mathrm{p}) + (1-\mathrm{y})\log(1-\mathrm{p})) \tag{5}$$

where:

- y is the sample label y∈{0,1};
- p is the probability of each sample belonging to a class, e.g., p = Pr(y = 1).

The precision metric is the ratio of the true positive samples over the total positive predictions made by the model and was calculated as shown in Equation (6), whilst the recall metric is the ratio of the true positives over the overall true predictions, as shown in Equation (7) [32]:

$$\mathrm{precision} = \frac{TP}{TP+FP} \tag{6}$$

$$\mathrm{recall} = \frac{TP}{TP+TN} \tag{7}$$

Lastly, the F1-score metric is the harmonic mean between precision and recall and was calculated using Equation (8) [34]:

$$\mathrm{F1-score} = 2 \times \frac{1}{\frac{1}{precision}+\frac{1}{recall}} \tag{8}$$

3.1. Imbalanced Dataset Results

The algorithms that used the imbalanced dataset of the 4950 records as input were the logistic regression, the neural network, the random forest and the penalized support vector classifier. Table 3 summarizes the accuracy, loss, precision, recall and F1-score metrics for each of the aforementioned algorithms.

Table 3. Algorithms' results using the imbalanced dataset as input.

Algorithm	Accuracy	Loss	Precision	Recall	F1-Score
LR	0.9980	0.0698	0.5000	0.5000	0.5000
NN	1.0000	0.0002	0.5000	0.5000	0.5000
RF	1.0000	0.0000	1.0000	1.0000	1.0000
SVC	0.9994	0.0209	0.8800	1.0000	0.9300

As is obvious from Table 3, all algorithms performed quite well in terms of the accuracy and loss metrics, whilst for precision, recall and F1-score, only the random forest and SVC had promising results. An interesting aspect of these results is the 0.5 values for precision, recall and F1-score for the LR and NN algorithms. These results were reached as these two algorithms predicted only the class labeled as '0'. Thus, due to the imbalanced nature of the dataset and the fact that the records labeled as '1' only total 10 out of the 4950 records, the accuracy and loss metrics were expected to be good. However, those two methods did not predict any sample with the label '1', and thus it can de deduced that they did not perform well with the provided dataset.

On the other hand, the random forest and SVC algorithms performed quite well, despite the fact that the dataset is imbalanced, confirming that the nature of these algorithms is more suitable for this type of problem.

3.2. Up-Sampled Dataset Results

Following the imbalanced dataset results for the logistic regression and neural network methods, their performance was also evaluated using an up-sampled dataset which was created as described in Section 2.2. The random forest and SVC methods were not evaluated, as described in Section 2.2, due to the fact that their purpose, in the context of this study, was to evaluate and prove their suitability when imbalanced datasets are used as input. Table 4 summarizes the accuracy, loss, precision, recall and F1-score metrics for each of the RF and NN models.

Table 4. Algorithms' results using the up-sampled dataset as input.

Algorithm	Accuracy	Loss	Precision	Recall	F1-Score
LR	0.9992	0.0276	1.0000	1.0000	1.0000
NN	1.0000	0.0000	0.3300	0.5000	0.4000

The up-sampled dataset significantly improved the results of the LR algorithm, as the accuracy and loss metrics were, again, quite promising and also the precision, recall and F1-score were equal to 1. These results reveal that the LR method is quite efficient for balanced datasets, confirming the need to employ an up-sampled balanced dataset.

However, the balanced dataset did not improve the performance of the NN. This result reveals that NNs are not directly dependent on the dataset balance (balanced or imbalanced) but rather rely on the size of the dataset. Thus, it is highly possible for a neural network to perform better for larger and more complex datasets, as they require more data in order to be trained and adjusted so as to achieve correct predictions.

3.3. Feature Importance Results

In addition to the metrics presented for both datasets before, it is worth examining the importance of each feature calculated for the methods presented in Section 2.2. The feature importance techniques provide useful insights concerning the usability of each of the five input attributes of the evaluated dataset in relation to their overall contribution to the predicted result(s). The assigned score on each target (input) feature reflects its importance and its role in the overall training and evaluation processes, as well as in the context of distinct ML algorithm implementation. This can lead to an overall improvement in the performance and the degree of effectiveness of the extracted predictive model, after a possible reduction in the dimensions of the initial dataset [35].

Figures 3–6 depict the feature importance of every algorithm that used the imbalanced dataset as input, namely, the logistic regression, neural network, random forest and penalized support vector classifier. Similarly, Figures 7 and 8 present the feature importance for the logistic regression and neural network when the up-sampled dataset was used as input.

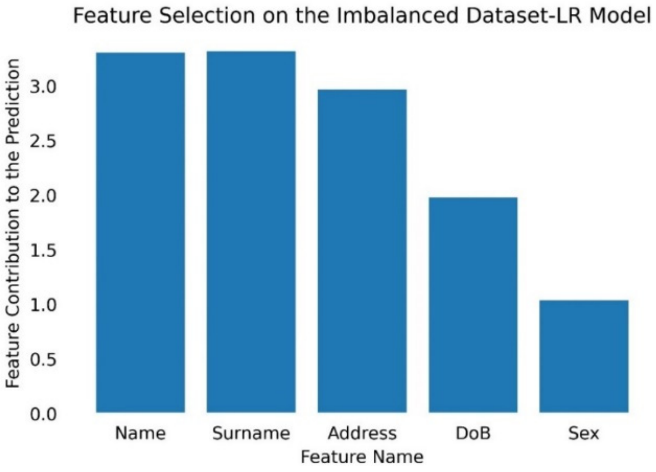

Figure 3. Feature importance of the LR model using the imbalanced dataset as input.

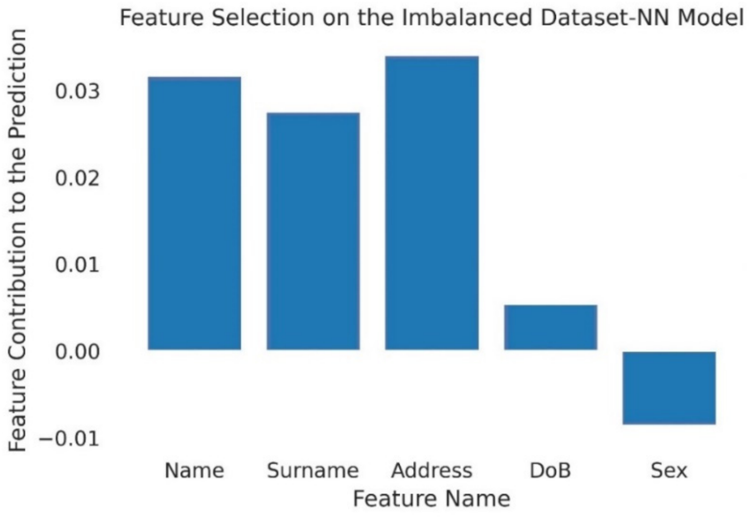

Figure 4. Feature importance of the NN model using the imbalanced dataset as input.

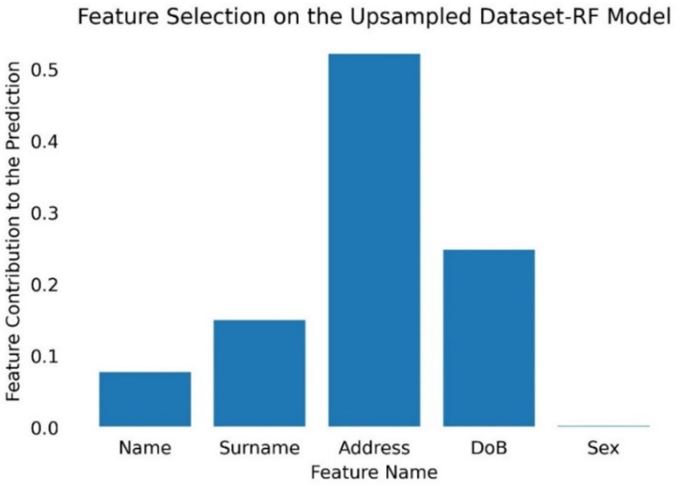

Figure 5. Feature importance of the RF model using the imbalanced dataset as input.

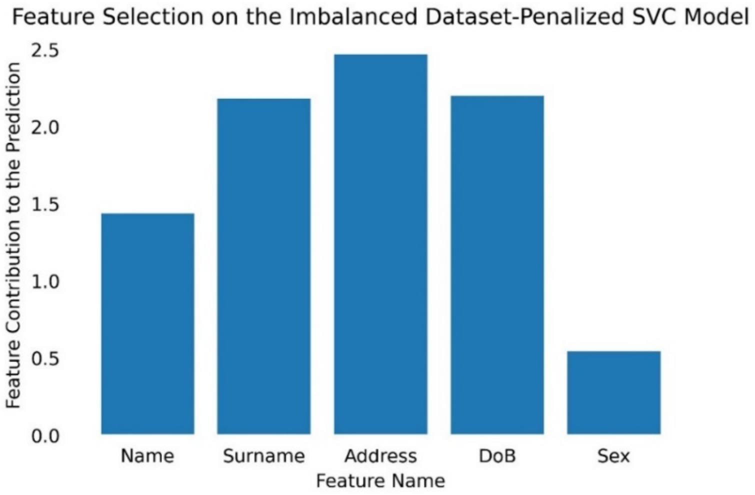

Figure 6. Feature importance of the SVC model using the imbalanced dataset as input.

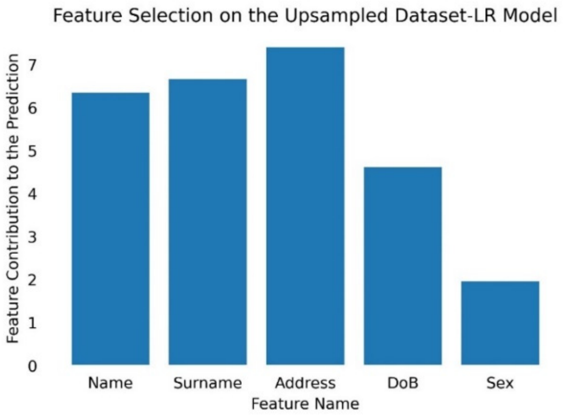

Figure 7. Feature importance of the LR model using the up-sampled dataset as input.

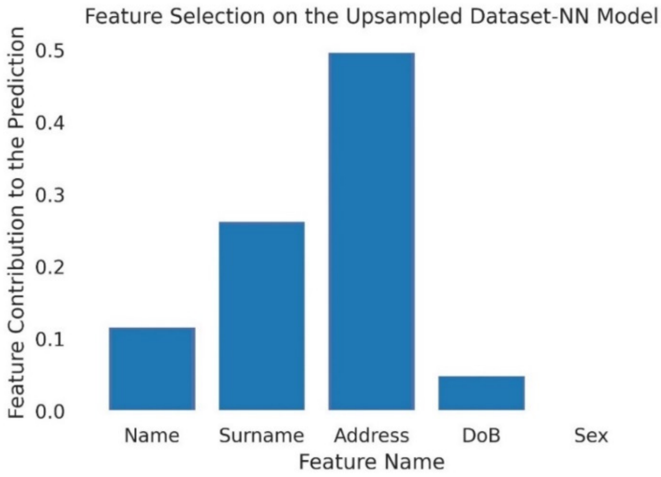

Figure 8. Feature importance of the NN model using the up-sampled dataset as input.

The first name (name) and surname features have the greatest impact in the LR method when the imbalanced dataset is used, whilst the sex attribute has the least impact when the imbalanced dataset is used as input (Figure 3). The LR method using the up-sampled dataset (Figure 7) indicates the address feature as the most important, whilst the sex attribute continues to be the least important. As far as the NN method is concerned, the address is the most important feature for both the imbalanced (Figure 4) and up-sampled (Figure 8) datasets. Additionally, the NN method deems the sex feature as the least important. The RF (Figure 5) and penalized SVC (Figure 6) methods both used the imbalanced dataset as input, and the feature importance follows a similar pattern in these two methods. More specifically, the address comes first, with the name, the surname and the date of birth (DoB) following, whilst the sex feature is again the least important.

4. Discussion

The current study expanded the findings of [18] considering the detection of duplicate person instances. The traditional string-matching methods have proven to be significantly time-consuming and lacking of computational efficiency. Thus, the exploration of ML- and DL-based techniques, in order to detect duplicate instances in data in a fast manner, gains ground. The main purpose of this study was to explore the capabilities of different ML and DL methods and their efficiency both for imbalanced and balanced datasets.

In Section 3, which is dedicated to the results of the methods examined, useful insights are provided into the different methods employed and their corresponding performance. First of all, the tree-based method random forest displayed very promising results using an imbalanced dataset, as did the penalized SVC method. Imbalanced datasets in terms of person instances are quite often used, since such datasets frequently contain a small proportion of duplicated records, and, in most cases, such duplicates are the result of mistakes such as typing errors or incomplete records.

On the other hand, the logistic regression method did not perform well with the imbalanced dataset, as it was unable to detect the rarely occurring similar persons. However, when the dataset was balanced using the up-sampling technique mentioned in Section 2, the LR method achieved remarkable results. Therefore, when the input dataset is balanced, the LR algorithm can be employed and trained to detect duplicates.

In addition to the ML methods discussed above, an NN was also evaluated. The NN results, both for the imbalanced and the up-sampled dataset, were not as promising as those extracted via other methods. This conclusion does not downgrade the potential and the effectiveness of neural networks but indicates that the size and the complexity of the dataset used in this study were not sufficient to train a neural network correctly. The size of a dataset, as well as its complexity and the feature selection, is crucial for effective neural network training [36]. As a matter of fact, this could be a future extension of this work, exploring the possibilities of a neural network given a different, more complex and larger dataset, alongside detailed research about the fine-tuning of neural networks by combining different sets of parameters.

It is worth mentioning that the results produced and presented for the purposes of this study in Section 3 are quite promising and indicative of the performance of ML and DL for different types of datasets (imbalanced and up-sampled). As can be seen from Section 1.2., where related works are presented, the results of this study also confirm the efficiency of ML models for duplication detection or data linkage. In addition to this, the results for the described methodology achieved performance metrics, in terms of accuracy, loss, recall, precision and F1-score, above 90%, despite the fact that the data tested and evaluated belong to different domains or have different formats.

Future research on content verification and duplication detection could engage larger and more complex datasets consisting of heterogeneous data. Additionally, through this research study, it becomes clear that a near-real-time solution for duplication detection and fusion could exist by engaging pretrained ML models. Research efforts should aim to explore solutions that will be agnostic of the type of data concerned, so as to offer accurate and timely results for vast volumes of data in several different domains.

5. Conclusions

In this study, a comparative analysis between different ML and DL methods for person instance verification was performed. Two different datasets, an imbalanced and a balanced one, were provided as input to different algorithms, using the up-sampling technique. Following the preprocessing and the resampling procedures, the logistic regression and neural network algorithms were benchmarked for both datasets, whilst the random forest and the penalized SVC were evaluated for the imbalanced dataset. In addition to the ML performance metrics (accuracy, loss, precision, recall and F1-score), a feature importance analysis of the datasets' features (first name, surname, address, DoB and sex) was also conducted.

The results reveal that the RF and penalized SVC algorithms performed well using the imbalanced dataset, whilst the LR algorithm returned promising results for the up-sampled dataset. The NN did not perform as well as the other models on either dataset, and this performance result mainly occurred due to the specific datasets' attributes.

Author Contributions: Conceptualization, K.D., E.A., T.A. and N.P.; methodology, N.P., T.A. and E.A.; software, N.P.; validation, N.P., T.A., E.A. and K.D.; formal analysis, T.A., E.A., K.D. and N.P.; investigation, N.P. and T.A.; resources, T.A. and N.P.; data curation, N.P. and T.A.; writing—original draft preparation, T.A. and N.P.; writing—review and editing, E.A., T.A., K.D. and N.P.; visualization, N.P. and T.A.; supervision, E.A. and K.D.; funding acquisition, K.D. and E.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was performed in the context of the ENSURESEC project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 883242. This paper reflects only the authors' views, and the Commission is not responsible for any use that may be made of the information it contains.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. CISCO. *The Zettabyte Era: Trends and Analysis*; e Cisco Visual Networking Index (Cisco VNI); CISCO: San Jose, CA, USA, 2016.
2. The World Bank. *Crossing Borders*; World Development Report; The World Bank: Washington, DC, USA, 2021.
3. Leonelli, S. Scientific Research and Big Data. In *The Stanford Encyclopedia of Philosophy*; Zalta, E.N., Ed.; Metaphysics Research Lab., Stanford University: Stanford, CA, USA, 2020.
4. Zhu, G.; Zhang, X.; Wang, L.; Zhu, Y.; Dong, X. An Intelligent Data De-Duplication Based Backup System. In Proceedings of the 2012 15th International Conference on Network-Based Information Systems, Melbourne, Australia, 26–28 September 2012; pp. 771–776.
5. Hall, D.L.; Llinas, J. An Introduction to Multisensor Data Fusion. *Proc. IEEE* **1997**, *85*, 6–23. [CrossRef]
6. Akter, S.; Wamba, S.F. Big Data Analytics in E-Commerce: A Systematic Review and Agenda for Future Research. *Electron. Mark.* **2016**, *26*, 173–194. [CrossRef]
7. Tran, M.-Q.; Elsisi, M.; Mahmoud, K.; Liu, M.-K.; Lehtonen, M.; Darwish, M.M.F. Experimental Setup for Online Fault Diagnosis of Induction Machines via Promising IoT and Machine Learning: Towards Industry 4.0 Empowerment. *IEEE Access* **2021**, *9*, 115429–115441. [CrossRef]
8. Ćirović, G.; Pamučar, D.; Božanić, D. Green Logistic Vehicle Routing Problem: Routing Light Delivery Vehicles in Urban Areas Using a Neuro-Fuzzy Model. *Expert Syst. Appl.* **2014**, *41*, 4245–4258. [CrossRef]
9. Policarpo, L.M.; da Silveira, D.E.; da Rosa Righi, R.; Stoffel, R.A.; da Costa, C.A.; Barbosa, J.L.V.; Scorsatto, R.; Arcot, T. Machine Learning through the Lens of E-Commerce Initiatives: An up-to-Date Systematic Literature Review. *Comput. Sci. Rev.* **2021**, *41*, 100414. [CrossRef]
10. Carvalho, M.; Laender, A.; Gonçalves, M.; Silva, A. A Genetic Programming Approach to Record Deduplication. *Knowl. Data Eng. IEEE Trans.* **2012**, *24*, 399–412. [CrossRef]
11. Christen, P.; Goiser, K. Towards Automated Data Linkage and Deduplication. *Computer* **2019**, *16*, 22–24.
12. Elfeky, M.G.; Verykios, V.S.; Elmagarmid, A.K. TAILOR: A Record Linkage Toolbox. In Proceedings of the 18th International Conference on Data Engineering, San Jose, CA, USA, 26 February–1 March 2002; pp. 17–28.
13. Gschwind, T.; Miksovic, C.; Minder, J.; Mirylenka, K.; Scotton, P. Fast Record Linkage for Company Entities. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 623–630.
14. Rajbahadur, G.K.; Wang, S.; Ansaldi, G.; Kamei, Y.; Hassan, A.E. The Impact of Feature Importance Methods on the Interpretation of Defect Classifiers. *IEEE Trans. Softw. Eng.* **2021**, 1. [CrossRef]
15. Zhu, Z.; Ong, Y.-S.; Dash, M. Wrapper–Filter Feature Selection Algorithm Using a Memetic Framework. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2007**, *37*, 70–76. [CrossRef] [PubMed]
16. Tran, M.-Q.; Elsisi, M.; Liu, M.-K. Effective Feature Selection with Fuzzy Entropy and Similarity Classifier for Chatter Vibration Diagnosis. *Measurement* **2021**, *184*, 109962. [CrossRef]
17. Niño-Adan, I.; Manjarres, D.; Landa-Torres, I.; Portillo, E. Feature Weighting Methods: A Review. *Expert Syst. Appl.* **2021**, *184*, 115424. [CrossRef]
18. Alexakis, T.; Peppes, N.; Adamopoulou, E.; Demestichas, K.; Remoundou, K. Evaluation of Content Fusion Algorithms for Large and Heterogeneous Datasets. In *Security Technologies and Social Implications: An European Perspective*; Wiley-IEEE Press (pending publication): Hoboken, NJ, USA, 2022.
19. Jaro, M.A. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *J. Am. Statitstical Assoc.* **1989**, *84*, 414–420. [CrossRef]
20. Winkler, W. *String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage*; ERIC: Middletown, OH, USA, 1990; pp. 354–359.
21. Levenshtein, V.I. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Sov. Phys. Dokl.* **1966**, *10*, 707–710.
22. Gomaa, W.H.; Fahmy, A.A. A Survey of Text Similarity Approaches. *Int. J. Comput. Appl.* **2013**, *68*, 13–18.
23. Jaccard, P. Distribution de La Flore Alpine Dans Le Bassin Des Dranses et Dans Quelques Régions Voisines. *Bull. Soc. Vaud. Sci. Nat.* **1901**, *37*, 241–272. [CrossRef]
24. Weisstein, E.W. Combination. Available online: https://mathworld.wolfram.com/Combination.html (accessed on 9 December 2021).

25. Marqués, A.I.; García, V.; Sánchez, J.S. On the Suitability of Resampling Techniques for the Class Imbalance Problem in Credit Scoring. *J. Oper. Res. Soc.* **2013**, *64*, 1060–1070. [CrossRef]
26. More, A. Survey of Resampling Techniques for Improving Classification Performance in Unbalanced Datasets. *arXiv* **2016**, arXiv:1608.06048.
27. Peppes, N.; Daskalakis, E.; Alexakis, T.; Adamopoulou, E.; Demestichas, K. Performance of Machine Learning-Based Multi-Model Voting Ensemble Methods for Network Threat Detection in Agriculture 4.0. *Sensors* **2021**, *21*, 7475. [CrossRef]
28. Islah, N.; Koerner, J.; Genov, R.; Valiante, T.A.; O'Leary, G. Machine Learning with Imbalanced EEG Datasets Using Outlier-Based Sampling. In Proceedings of the 2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC), Montreal, QC, Canada, 20–24 July 2020; pp. 112–115.
29. Maldonado, S.; López, J. Dealing with High-Dimensional Class-Imbalanced Datasets: Embedded Feature Selection for SVM Classification. *Appl. Soft Comput.* **2018**, *67*, 94–105. [CrossRef]
30. Ganganwar, V. An Overview of Classification Algorithms for Imbalanced Datasets. *Int. J. Emerg. Technol. Adv. Eng.* **2012**, *2*, 42–47.
31. Panigrahi, R.; Borah, S.; Bhoi, A.K.; Ijaz, M.F.; Pramanik, M.; Kumar, Y.; Jhaveri, R.H. A Consolidated Decision Tree-Based Intrusion Detection System for Binary and Multiclass Imbalanced Datasets. *Mathematics* **2021**, *9*, 751. [CrossRef]
32. Hossin, M.; Sulaiman, M.N. A Review on Evaluation Metrics for Data Classification Evaluations. *Int. J. Data Min. Knowl. Manag. Process* **2015**, *5*, 1.
33. Bishop, C.M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*; Springer: Berlin/Heidelberg, Germany, 2006; ISBN 0-387-31073-8.
34. Mishra, A. Metrics to Evaluate Your Machine Learning Algorithm. Available online: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234 (accessed on 9 December 2021).
35. Zien, A.; Krämer, N.; Sonnenburg, S.; Rätsch, G. The Feature Importance Ranking Measure. In *Proceedings of the Machine Learning and Knowledge Discovery in Databases*; Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 694–709.
36. Alwosheel, A.; van Cranenburgh, S.; Chorus, C.G. Is Your Dataset Big Enough? Sample Size Requirements When Using Artificial Neural Networks for Discrete Choice Analysis. *J. Choice Model.* **2018**, *28*, 167–182. [CrossRef]