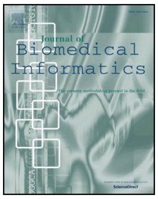
Journal of Biomedical Informatics 95 (2019) 103220

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin



Statistical supervised meta-ensemble algorithm for medical record linkage

Kha Vo[a,b], Jitendra Jonnagaddala[a,c,*], Siaw-Teng Liaw[a,c]

[a] School of Public Health and Community Medicine, UNSW Sydney, Australia
[b] School of Electrical and Data Engineering, Faculty of Electrical and Information Technology, University of Technology Sydney, Australia
[c] WHO Collaborating Centre for eHealth, UNSW Sydney, Australia

ABSTRACT

Identifying unique patients across multiple care facilities or services is a major challenge in providing continuous care and undertaking health research. Identifying and linking patients without compromising privacy and security is an emerging issue in the big data era. The large quantity and complexity of the patient data emphasize the need for effective linkage methods that are both scalable and accurate. In this study, we aim to develop and evaluate an ensemble classification method using the three most typically used supervised learning methods, namely support vector machines, logistic regression and standard feed-forward neural networks, to link records that belong to the same patient across multiple service locations. Our ensemble method is the combination of bagging and stacking. Each base learner's critical hyperparameters were selected through grid search technique. Two synthetic datasets were used in this study namely FEBRL and ePBRN. ePBRN linkage dataset was based on linkage errors noticed in the Australian primary care setting. The overall linkage performance was determined by assessing the blocking performance and classification performance. Our ensemble method outperformed the base learners in all evaluation metrics on one dataset. More specifically, the precision, which is average of individual precision scores in case of base learners increased from 90.70% to 94.85% in FEBRL, and from 62.17% to 99.28% in ePBRN. Similarly, the F-score increased from 94.92% to 98.18% in FEBRL, and from 72.99% to 91.72% in ePBRN. Our experiments suggest that we can significantly improve the linkage performance of individual algorithms by employing ensemble strategies.

1. Introduction

Data record linkage is the task of matching duplicate records that belong to the same entity within a dataset or across different datasets. For instance, a patient admitted into a hospital multiple times (or at different hospitals) might often possess two or more different identification records within the same database or across different databases. These identification records that belong to the same person should be linked together. It is also important to link other patient-related data, such as diagnoses and treatment information, to provide better healthcare services. Record linkage concepts were also applied in non-health-related fields, such as online customer profile for targeted online marketing [1].

Various approaches have been applied to data linkage and de-duplication. Machine learning methods include supervised methods such as NCLink [2], HARRA [3], and MFIBlock [4], and unsupervised methods such as canopy clustering [5]. In an another study, hierarchical linkage clustering method was proposed to maintain linear complexity, thereby allowing to process large amount of data [2]. A fast-iterative record linkage algorithm based on reusable hash table that exploits various characteristics of the data was also presented [3]. In a study by Sagi et al. [4] decision-tree based approach was used to identify possible linkable records. Apache Drill was also used by researchers to develop a linkage platform for patient-related data [6]. Furthermore, a few studies reviewed and compared various linkage methods [7–12].

In record data linkage, standardization and normalization of text-based attributes such as addresses, and names is a fundamental step. A probabilistic approach, based on hidden Markov models, can be used to process textual attributes into different formats (names, addresses) to standardize these attributes before feeding them into the classification model [13]. Phonetic encoding was employed in a few studies to link patients in population health datasets [14,15]. Blocking is another important step in record linkage, which can significantly reduce the number of suspicious pairs of records for classification. In the blocking phase, suspicious match pairs are generated based on heuristics. For example, any two records with similarity score that is based on a text field such as surname, higher than a pre-specified threshold will be considered for linkage. For large amounts of data, a meta-blocking algorithm based on the MapReduce can also be employed [16]. When there is no training data available, unsupervised blocking technique such as clustering can be used [5].

A challenging aspect in designing and developing record linkage methods is the lack of real-world datasets. Researchers often generate and use synthetic datasets to evaluate linkage methods. Several tools and packages can be used to generate synthetic datasets as well as to

[*] Corresponding author.
E-mail address: z3339253@unsw.edu.au (J. Jonnagaddala).

https://doi.org/10.1016/j.jbi.2019.103220
Received 14 December 2018; Received in revised form 23 April 2019; Accepted 28 May 2019
Available online 31 May 2019
1532-0464/

perform record linkage. For example, FRIL [10] allows users to specify various linkage attributes, blocking functions, and classification models. Similarly, FEBRL [8,9] offers users a wide range of blocking techniques. These include neighbor indexing and user-defined feature indexing. FEBRL is also capable of performing phonetic encoding, extracting features and deploying various machine learning algorithms. The German Record Linkage Center provided TDGen [17] to generate user-defined linkage errors, based on spellings. These error rates are further used to generate synthetic duplicate records from an original dataset, and subsequently combine both the original and duplicate datasets to form a full dataset for evaluating linkage methods. However, synthetic datasets are subjected to various limitations. A reliable synthetic dataset requires significant manual efforts in annotating each suspicious linkage pair. As a result, the linkage error estimates generated based on limited manually annotated records may be subjected to bias.

Most of the studies discussed above, typically use single machine learning algorithm or model, to perform medical record linkage. To our knowledge, we are not aware of any studies on medical record linkage that employ ensemble-based methods. Additionally, we are also not aware of any studies that have generated synthetic dataset based on real primary care patient data. In this study, we address these research gaps by (i) proposing an ensemble-based supervised learning strategy that exploits the strengths of each individual base learner (ii) employing state-of-the-art feature engineering and blocking techniques and (iii) generating a synthetic record linkage dataset based on the linkage error statistics observed in Australian primary care setting.

2. Methods

The proposed ensemble linkage framework consists of five consecutive processing steps, namely data cleaning and standardization, blocking, feature engineering, ensemble learning (or classification), and evaluation (Fig. 1). First, the original raw dataset was cleaned and standardized. Next, the blocking step extracted suspicious candidate pairs. Subsequently, the numerical features were engineered based on all data attributes (column fields) [4,5,13]. For simplicity, only two features were calculated (the soft scores of similarities of name, and of date of birth). Next, all the feature vectors representing suspicious candidate pairs were fed into a trained ensemble model to generate the output labels (matched or non-matched). Finally, an evaluation step was conducted to measure the linkage classification performance. The meta-ensemble code and data used in this study is available at https://github.com/ePBRN/Medical-Record-Linkage-Ensemble/.

2.1. Data cleaning and standardization

Data cleaning and standardization is an important processing step as the quality affects the subsequent linkage classification performance significantly. The specific steps undertaken in this study are explained below.

(1) All of the patients' textual data were converted into lower case and split into separate tokens. In many records, we found that name tokens are often swapped within one field or between fields. The benefit of this tokenization process is therefore to detect swapped words in such cases. This tokenization process can also handle many names that are not split by a space but by special symbols (e.g. commas or hyphens). With tokenized names, the feature engineering step is facilitated substantially. Tokenization, potentially, has a high impact on the descriptions of health data by recognizing the long prescription dosage in number.

(2) An extra preprocessing step was employed to correct names that are contaminated with punctuations, periods, or non-alphabetical symbols owing to human typographical errors.

(3) Specific standardization techniques were conducted depending on the characteristics of each field. Dates of birth were parsed to separate fields (i.e., date, month, year). A swap between day and month is performed if the value of month is greater than 12. The standardization of address field faced different challenges as compared to the standardization of dates of birth, because addresses contain more separate tokens as compared to three tokens of dates of birth. Address tokens were split into predefined categories (i.e., street number, street name, street type, suburb, city) by some heuristics. For example, we first searched for a textual token that is matched with our dictionary of street types (i.e., "st", "rd," "ave," corresponding to "street," "road," and "avenue", respectively) to put them in the street type field; subsequently, any textual token followed by the street type token is categorized as a street name.

2.2. Blocking criteria

The number of possible candidate pairs for linkage increases exponentially with the number of records or patients in the dataset. For instance, a dataset with 50,000 records may possibly produce over 2 billion pairs to compare. It is crucial to eliminate pairs which have low probability of belonging to the same patient using blocking [11,12]. We employed the nearest neighbor clustering based blocking criteria in this study [11]. Though the goal of the blocking criteria is to reduce computational complexities, sometimes it may have undesirable effects. For

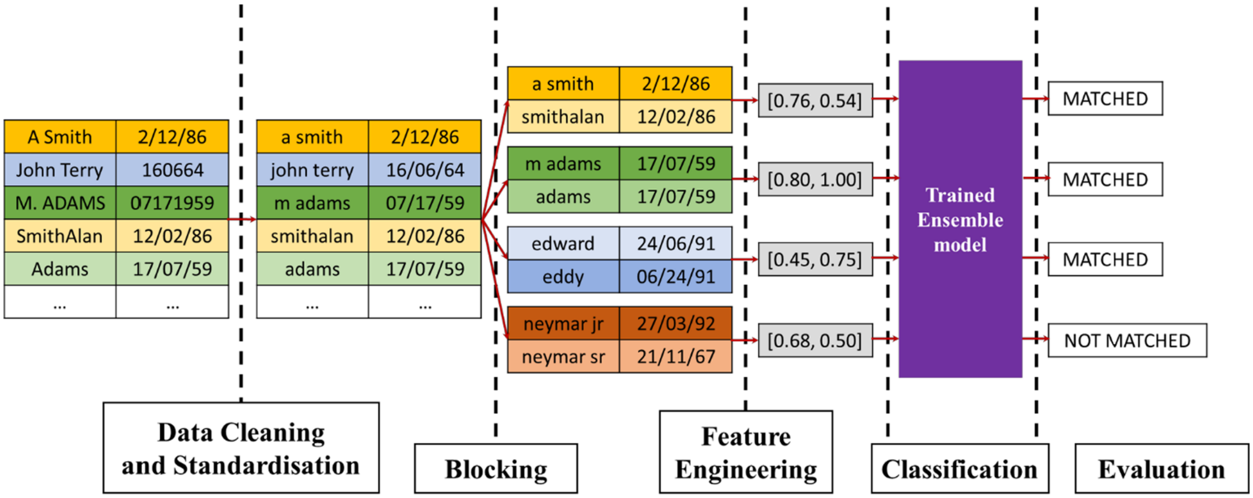


Fig. 1. Proposed ensemble based medical record linkage framework.

Table 1
List of features for classification.

Feature description	No. of features
Feature set 1	
Jaro–Winkler similarity of given names or surnames	2
Jaro–Winkler similarity of Soundex-encoded given names or surnames	2
Jaro–Winkler similarity of Nysiis-encoded given names or surnames	2
Levenshtein similarity of two address fields	2
Exact street number, postcode, day, month, year of birth	5
Total	13
Feature set 2	
Levenshtein similarity of given names, surnames	2
Jaro–Winkler similarity of given names, surnames, postcode	3
Exact compare of postcode and three address fields	4
Check if surname and given name are swapped	1
Check if date and month of birth are swapped	1
Check if date and month of birth are reset to default (01/01)	1
Check if surname and given name are joined into one field	1
Check if surname and given name are join with dashed into one field	1
Check if surname is abbreviated by the first letter	1
Total	15

example, in situations where a patient changes surname, the Levenshtein distance of similarity, will be extremely large and may lead to undesirable drop of this candidate pair.

2.3. Feature engineering

Supervised and unsupervised record linkage methods rely heavily on hand-crafted features. A set of well-engineered features may be able to generalize the model to handle complex scenarios. For example, good feature engineering might be able to distinguish twins with same surnames and dates of birth. Each feature represents a certain level of similarity between the corresponding attributes of two records in a candidate pair [18]. We generated various features for both textual and numerical attributes (Table 1). For textual attributes, we performed character-level phonetic-encoding features on people and regional names, and token-level phonetic-encoding features on long addresses. The Jaro–Winkler and Levenshtein similarity, as well as Soundex and Nysiis encoding, are described in detail elsewhere [19].

2.4. Ensemble learning

Ensemble learning is to combine different classification or regression models in a specific manner to boost performance [20–22]. In this study, we applied bagging and stacking.

2.4.1. Bagging

The key idea of bagging is to obtain a lower generalization error by averaging models that were trained on different subsets of the full dataset [22,23]. Bagging learners have been shown to outperform [20,22].

Suppose we draw each point $(y, \boldsymbol{x})$ from the probability distribution P to form a training set $\mathcal{L}$, where y is the scalar of the class label and $\boldsymbol{x}$ is the feature vector. Let $f(\boldsymbol{x}, \mathcal{L})$ be the learner that is trained on $\mathcal{L}$; subsequently, the ensemble bagging learner averaged on all possibilities of $\mathcal{L}$ can be defined as follows:

$$f_{\text{bagging}}(\boldsymbol{x}, P) = \mathbb{E}_{\mathcal{L}}[f(\boldsymbol{x}, \mathcal{L})]. \tag{1}$$

Let us denote $\boldsymbol{X}, Y$ as the random variables independent of $\mathcal{L}$ and with distribution P. We calculate the average prediction error e on the weak learning $f(\boldsymbol{x}, \mathcal{L})$ over all possibilities of $\mathcal{L}$ by

$$\begin{aligned} \mathrm{e} &= \mathbb{E}_{\mathcal{L}}\mathbb{E}_{\boldsymbol{X},Y}[Y - f(\boldsymbol{X}, \mathcal{L})]^2 \\ &= \mathbb{E}_{\mathcal{L}}\mathbb{E}_{\boldsymbol{X},Y}[Y^2] - 2\mathbb{E}_{\boldsymbol{X},Y}[Y]\mathbb{E}_{\mathcal{L}}[f(\boldsymbol{X}, \mathcal{L})] + \mathbb{E}_{\boldsymbol{X},Y}\mathbb{E}_{\mathcal{L}}[f(\boldsymbol{X}, \mathcal{L})]^2 \\ &= \mathbb{E}_{\boldsymbol{X},Y}[Y^2] - 2\mathbb{E}_{\boldsymbol{X},Y}[Yf_{\text{bagging}}] + \mathbb{E}_{\boldsymbol{X},Y}\mathbb{E}_{\mathcal{L}}[f(\boldsymbol{X}, \mathcal{L})]^2. \end{aligned} \tag{2}$$

Meanwhile, the error produced by f_{bagging} is

$$\begin{aligned} e_{\text{bagging}} &= \mathbb{E}_{\boldsymbol{X},Y}[Y - f_{\text{bagging}}(\boldsymbol{X}, P)]^2 \\ &= \mathbb{E}_{\mathbf{X},Y}[Y^2] - 2\mathbb{E}_{\boldsymbol{X},Y}[Yf_{\text{bagging}}] + \mathbb{E}_{\boldsymbol{X},Y}[\mathbb{E}_{\mathcal{L}}f(\boldsymbol{X}, \mathcal{L})]^2. \end{aligned} \tag{3}$$

Because $\mathbb{E}Z^2 \geq \mathbb{E}^2 Z$ for all Z we obtain $e \geq e_{\text{bagging}}$, and the margin between e and e_{bagging} depends on the variance of $f_{\text{bagging}}(\boldsymbol{X}, P)$ as follows:

$$\begin{aligned} e - e_{\text{bagging}} &\geq \mathbb{E}_{\boldsymbol{X},Y}[\mathbb{E}_{\mathcal{L}}[f(\boldsymbol{X}, \mathcal{L})]^2 - \mathbb{E}_{\mathbf{X},Y}[\mathbb{E}^2_{\mathcal{L}}f(\boldsymbol{X}, \mathcal{L})] \\ &= \mathbb{E}_{\boldsymbol{X},Y}[\mathrm{Var}_{\mathcal{L}}f(\boldsymbol{X}, \mathcal{L})]. \end{aligned} \tag{4}$$

Hence, we expect to obtain a good bagging ensemble by drawing and training diverse subsets $\mathcal{L}$ from P. This was accomplished by splitting the originally provided training set into k separate non-overlapping *folds*, all of which contain the equal number of training samples. Each fold of the partition is validated by the learner trained on the samples from the other folds. The step-by-step bagging pseudo-code employed in this study is presented in Algorithm 1.

```
Algorithm 1: Bagging Algorithm
        Input: Training set $\mathcal{L} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$,
          Number of subsets $K$,
          Base learner algorithm $F$, dataset as input, output a classifier.
    Draw $K$ subsets with replacement from $\mathcal{L}$, denoted by $\mathcal{L}_k$.
    for $k = 1, \cdots, K$ do
        $f_k \leftarrow F(\mathcal{L}_k)$
    end for
    Output: Bagging classifier $f_{\text{bagging}}(\boldsymbol{x}) = \mathrm{argmax}_y \sum_{k=1}^{K} \mathbf{I}(f_k(\boldsymbol{x}) = y)$, where
    $\mathbf{I}(A) = 1$ if $A$ is true, and $\mathbf{I}(A) = 0$ if $A$ is false.
```

2.4.2. Stacking

Another powerful ensemble method is *stacking*, where the results from various learners trained on the same dataset are combined using another learner. We herein define the first level of learners as the *base learners*, and the learner that combined the base learners as the *stack learner*. We employed linear regression as the stacking ensemble learner because of its simplicity and robustness on low-dimensional regression problems [24]. Stacking is expected to generalize well when its base learners include both deterministic methods (SVMs, NNs) and probabilistic methods (LG, random forests) [20].

Suppose each of the N base learners is trained on the training set $\mathcal{L} = (\boldsymbol{x}_m, y_m)_{\mathrm{m}=1}^{|\mathcal{L}|}$, where $|\mathcal{L}|$ is the *size* of $\mathcal{L}$, i.e., the number of samples in $\mathcal{L}$. We denote the prediction on sample $\boldsymbol{x}$ of each base learner that is trained on set $\mathcal{L}$ as $f_n^{(\mathcal{L})}(\boldsymbol{x})$, $n = 1 \cdots N$. Stacking ensemble $f_{\text{stack}}(\boldsymbol{x})$ combines the predictions from all base learners by assigning non-negative weights to them.

$$f_{stack}(\boldsymbol{x}) = \sum_{n=1}^{N} \alpha_n f_n^{(L)}(\boldsymbol{x}),\ \alpha_n \geq 0. \tag{5}$$

The empirical squared prediction error e_{stack} of f_{stack} on $\mathcal{L}$ is

$$\begin{aligned} e_{stack} &= \textstyle\sum_m \left(y_m - \sum_n \alpha_n f_n^{(L)}(\boldsymbol{x}_{\boldsymbol{m}})\right)^2 \\ &= \textstyle\sum_i \sum_j \alpha_i \alpha_j R_{ij}, \end{aligned} \tag{6}$$

where R_{ij} is the *residual* error products of learner f_i and f_j.

$$R_{ij} = \sum_m (y_m - f_i^{(\mathcal{L})}(\boldsymbol{x}_m))(y_m - f_j^{(\mathcal{L})}(\boldsymbol{x}_m)). \tag{7}$$

Breiman et al. [21], using the duality technique in optimization theory, proved that any base learner n performs as only a stacked learner at best if and only if the following condition holds:

$$R_{nn} \leq R_{ni}, \quad \forall\ i. \tag{8}$$

From a statistical viewpoint, the condition above relates to the correlation $\rho_{A,B}$ between two random variables A and B as follows:

$$\rho_{A,B} = \frac{\mathbb{E}\,[(A - \mathbb{E}\,[A])(B - \mathbb{E}\,[B])]}{\sigma_A \sigma_B}, \tag{9}$$

where σ_A, σ_B are the standard deviations of A and B, respectively. Considering the residual error, $y_m - f_i^{(\mathcal{L})}(\boldsymbol{x}_m)$ of any base learner i as a random variable, it has a zero mean and $\rho_{i,i} = 1$ for all i. The correlation between the residuals of two base learners i, j is computed by

$$\rho_{i,j} = \frac{\sum_m (y_m - f_i^{(\mathcal{L})}(\boldsymbol{x}_m))(y_m - f_j^{(\mathcal{L})}(\boldsymbol{x}_m))}{\sqrt{\sum_m (y_m - f_j^{(\mathcal{L})}(\boldsymbol{x}_m))^2 \sum_m (y_m - f_j^{(\mathcal{L})}(\boldsymbol{x}_m))^2}}. \tag{10}$$

Using Eq. (10), we obtain the equivalent form of the condition in Eq. (8) as

$$\frac{\sigma_i}{\sigma_n} \leq \rho_{n,i}. \tag{11}$$

Intuitively, suppose we have a base learner i that is completely different from the best performing base learner n. If i produces comparable squared errors as those of n, then their correlation is approximately one, implying that learner i must be nearly identical to learner n. This yields the conclusion that no base learner can perform better than the stacked learner, if the base learners are diverse with comparable errors. The step-by-step pseudo-code for the two-level stacking algorithm employed in this study is presented in Algorithm 2.

Algorithm 2: Two-level Stacking Algorithm.

```
Input: Training set $\mathcal{L} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$,
    $N$ base learner algorithms $F_n$,
    1 meta learner algorithm $F$.
for $n = 1, \cdots, N$ do
    $f_n \leftarrow F_n(\mathcal{L})$
end for
Initialize second-level dataset$D \leftarrow \varnothing$
for $i = 1, \cdots, m$ do
  for $n = 1, \cdots, N$ do
      $z_{in} = f_n(x_i)$
  end for
  $D \leftarrow D \cup \{[z_{i1}, z_{i2}, \cdots, z_{iN}], y_i\}$
end for
Train meta model $f = F(D)$
Output: Stacking model $f_{\text{stack}}(\boldsymbol{x}) = f([f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_N(\boldsymbol{x})])$.
```

2.4.3. Combining bagging and stacking

An issue in the stacking ensemble is that $f_i^{(\mathcal{L})}(\boldsymbol{x}_m)$ will estimate back on $\boldsymbol{x}_m$ after it is used as a training sample, i.e., $\boldsymbol{x}_m$ is included in $\mathcal{L}$. Consequently, overfitting will occur and we cannot eliminate this bias [20,23]. In other words, in the first training stage, all the training samples serve to optimize the loss function, causing the resulting model to overfit. As a result, one must therefore divide the original training set into two subsets: one for training and one for validating, and use the validated scores as inputs for the stacking ensemble. Because the validated scores only constitute part of the original training set, different partitions are required where the union of validation sets of all partitions is the same as the original training set. This method is identical to the bagging ensemble.

Let us divide the training set $\mathcal{L}$ into K non-overlapping folds, denoted by $\mathcal{L}_k$, $k = 1, \cdots, K$. The number of samples in each fold is nearly equal, and the folds must cover the entire set, i.e., $\bigcup_k \mathcal{L}_k = \mathcal{L}$. Each fold k is validated by a base learner n that is trained on the samples combined from all the other folds, denoted by $\bar{\mathcal{L}}_k = \bigcup_{i \neq k} \mathcal{L}_i$. By further denoting the validated scores of the base learner n on fold k as $\mathbf{s}_{\text{nk}}$, we obtain the following $K \times N$ sets of scores:

$$\boldsymbol{s}_{nk} = f_n^{(\bar{\mathcal{L}}_k)}(\mathcal{L}_k) = [s_{nk,1} \quad s_{nk,2} \quad \cdots \quad s_{nk,|\mathcal{L}_k|}]. \tag{12}$$

By treating $\boldsymbol{s}_{nk}$ as a row vector, we can concatenate all the entry-level individual scores into a matrix $\mathbf{S}$ of size $|\mathcal{L}| \times N$ as follows:

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{s}_{11} & \boldsymbol{s}_{12} & \cdots & \boldsymbol{s}_{1K} \\ \boldsymbol{s}_{21} & \boldsymbol{s}_{22} & \cdots & \boldsymbol{s}_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{s}_{N1} & \boldsymbol{s}_{N2} & \cdots & \boldsymbol{s}_{NK} \end{bmatrix} = [\boldsymbol{c}_1 \quad \boldsymbol{c}_2 \quad \cdots \quad \boldsymbol{c}_{|\mathcal{L}|}]. \tag{13}$$

Each column of $\boldsymbol{S}$, denoted by $\boldsymbol{c}_m$,$m = 1, \cdots, |\mathcal{L}$ is the combination of scores of all base learners on one specific sample of $\mathcal{L}$. The stacking ensemble is therefore learned from the set of $|\mathcal{L}|$ vectors of dimension N, denoted by $\mathcal{L}_{\text{stack}}$ as

$$\mathcal{L}_{\text{stack}} = \{(\boldsymbol{c}_m, y_m)\}_{m=1}^{|\mathcal{L}|}. \tag{14}$$

2.5. Base learners

We employed three types of base learners for our ensemble (bagging–stacking) classifier, namely support vector machines (SVM), logistic regression (LR), and feed-forward neural networks (NN). SVM is a deterministic binary classification algorithm proposed by Vapnik et al. in 1995 [27]. Since its advent, SVM has gained significant popularity owing to its outstanding generalization capability on binary classification problems. One advantage of SVM is that the model solution only depends on a subset of the training data, called support vectors. These vectors are determined as the closest points to the decision boundary. Meanwhile, LR is a probabilistic approach that approximates the distribution of training samples on the feature space [28]. LR attempts to maximize the posterior class probability [28], while SVM attempts to maximize the margin between the closest support vectors [27]. As a result, SVM can find a solution which is as far as possible for the two classes, which is a property that LR does not offer. However, the simpler approach of LR is less prone to overfitting than SVM [27]. The completely different natures of SVM and LR yield some discrepancies on their class probabilities, which is useful for our ensemble method. A NN has been theoretically proven capable of constructing a model that fits any arbitrary distribution [29]. With SVM and LR, however, this objective cannot be achieved. Hence, a general NN typically performs better than SVM and LR with a tradeoff in training time. The NN used in this study is of fully connected feed-forward architecture with 256 units in the hidden layer followed by ReLu activation functions [30].

2.6. Hyperparameter selection

Choosing optimal hyperparameters for base learners is critical for high classification performance. The critical hyperparameters that were tuned included the generalization coefficient C for SVM, the inverse of

Table 2
Errors generated in the ePBRN synthetic dataset.

Error description	Example
Abbreviation on name fields	Michael → M
Join surname and given name into 1 field by dash	John Peter → John–Peter
Join surname and given name into 1 field by blank	John Peter → John Peter
Missing data	Peter→
Drop n last character in text fields	Peter → Pete
Swap surname and given name	John Peter → Peter John
Swap two random consecutive characters in text fields	Peter → Petre
Swap day and month fields	24/06 → 06/24
Reset day and/or month	24/06/1991 → 01/01/1991
Change year of birth by a specified margin	1980 → 1990
Drop leading zeros from day and/or month of birth	24/06 → 24/6
Change the whole address	14 Park Rd → 27 Sugar St.
Change any number of digits from zip code	2044 → 2144
Change the whole given name or surname	Mary Ward → Mary Johnson

Table 3
Classification schemes.

Scheme		Training set	Test set	Dataset	Feature set
A	Total records	5000	10,000	FEBRL	1
	Matched pairs	1165	5000		
B	Total records	14,078	11,731	ePBRN	2
	Matched pairs	3192	2653		

Table 4
Blocking performance based on different criteria. The metrics used are number of candidates (*nc*), pair completeness (*pc*), and reduction ratio (*rr*) [31].

Blocking Criterion	Measure	Scheme A	Scheme B
Surname	*nc*	170,843	33,832
	pc	66.50%	55.79%
	rr	99.65%	99.95%
Given name	*nc*	154,898	252,552
	pc	65.74%	59.07%
	rr	99.69%	99.63%
Postcode	*nc*	53,197	79,940
	pc	84.38%	92.80%
	rr	99.89%	99.88%
All	*nc*	372,073	362,910
	pc	97.88%	98.63%
	rr	99.26%	99.47%

regularisation strength C for LR, and the L2 regularization term parameter α for NN. These hyperparameters, which serve as regularization terms in each base learner, were vital and must be tuned prior to classification. We have employed grid search technique to identify best hyperparameters. The tuning values used were [.001, .002, .005, .01, .02, .05, .1, .2, .5, 1, 5, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000]. The search range for the hyperparameters was in logarithmic scale, which can better capture significant changes in the performance [27]. Additionally, we manually tuned other hyperparameters, such as regularisation types (L1, L2) in LR, activation functions (ReLu, logistic) in NN, and kernel types (linear, RBF) in SVM manually based on multiple experiments. For the stacked learner, as we employed a simple linear regressor $\mathbf{A}\mathbf{x} = \mathrm{b}$ on stacked features with equal coefficients, i.e. **A** is the identity matrix, we also performed manual search to tune the classification threshold b.

2.7. Datasets

The freely extensible biomedical record linkage (FEBRL) package was distributed with four datasets generated with a developed error generator [8]. We selected two non-overlapping sets, one for training and one for testing, as shown in Table 3. These datasets are of different sizes but have the same error types, primarily based on human typing errors. In order to overcome this, we generated a new synthetic dataset. We used data from electronic practice-based research network (ePBRN) data repository (Appendix A) to identify error estimates and replicated these errors on the FEBRL datasets [25,26]. We manually detected and labeled a subset of the 24,000 records in the ePBRN data repository. The percentage of records where there was linkage between two-records (22.75%), three-records (2.05%), and four-records (0.05%) was obtained. We used this information to incorporate the same into the FEBRL datasets. We randomly selected a subset of the FEBRL records to generate new synthetic duplicates. In that selected subset, we generated a duplicate matched record for each original record based on a collection of different error types with equal weights, as described in detail in Table 2. Each record in the original subset would subsequently be duplicated with the number of errors that is sampled from a Poisson distribution with an expectation of one and size of one. Subsequently, the type of error for that single record was chosen randomly from all error types presented in Table 2. Unlike some state-of-the-art data linkage synthetic generators, our generator can create some case-based matched links, such as twins (records only differing in given names), or married women (records differing in surname and living address).

We implemented two classification schemes based on the FEBRL and ePBRN datasets (Table 3). Each dataset included a training and a test set with features extracted described in the previous section.

3. Results

We summarized the performance of blocking in Table 4. Scheme A has a total of 49,995,000 possible pairs with 5000 matched pairs. Scheme B has a total of 68,802,315 possible pairs with 1716 matched pairs. For the assessment of blocking performance, we employed *pair completeness (pc) and* reduction ratio (rr).

The tuning performance for each hyperparameter combination is depicted in Fig. 2. For each base learner, only the best performing hyperparameter is selected for the subsequent ensemble method, which is shown in Table 5.

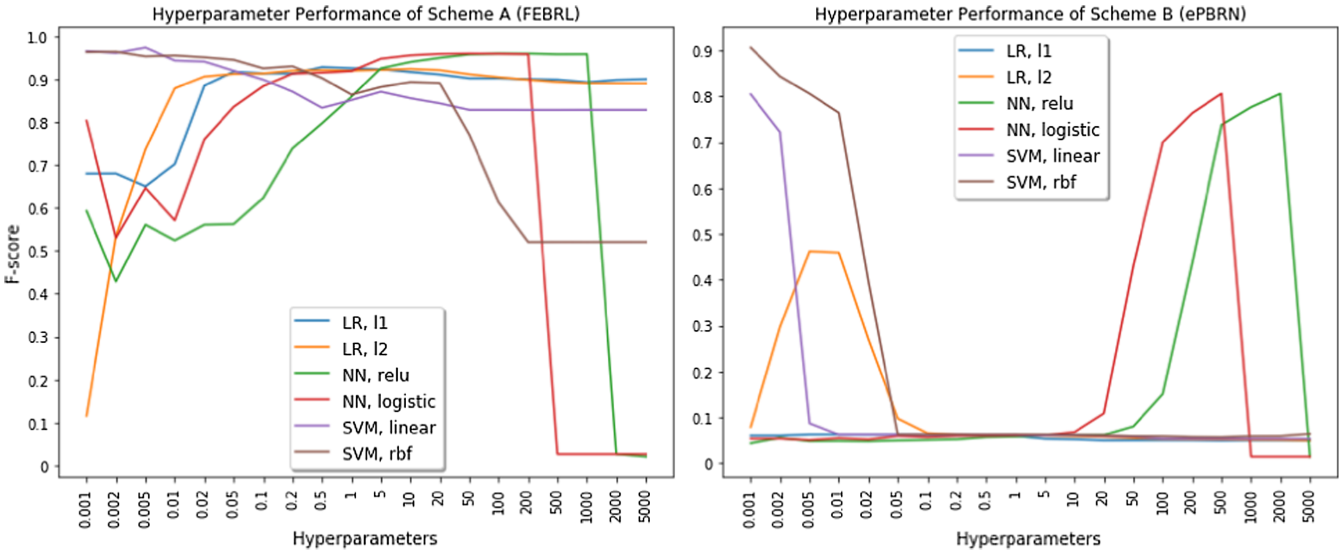


Fig. 2. The hyperparameters is the generalization coefficient C for SVM [27], the regularization parameter α for NN [30], and the inverse of regularisation C for LR [28].

Table 5
Hyperparameter selection for base learners using grid search.

Base learner	Scheme	Hyperparameter	fs(%)
SVM	A	Linear kernel with $C = 0.005$	97.23
SVM	B	RBF kernel with $C = 0.001$	90.67
NN	A	ReLu activation with $\alpha = 100$	96.08
NN	B	ReLu activation with $\alpha = 2000$	80.51
LR	A	Regularization l_2 with $C = 0.2$	91.44
LR	B	Regularization l_2 with $C = 0.005$	47.78

Table 6
Ensemble learning classification results. The metrics used in this table are precision (pr), recall (re), F-score (fs), and false counts (fc).

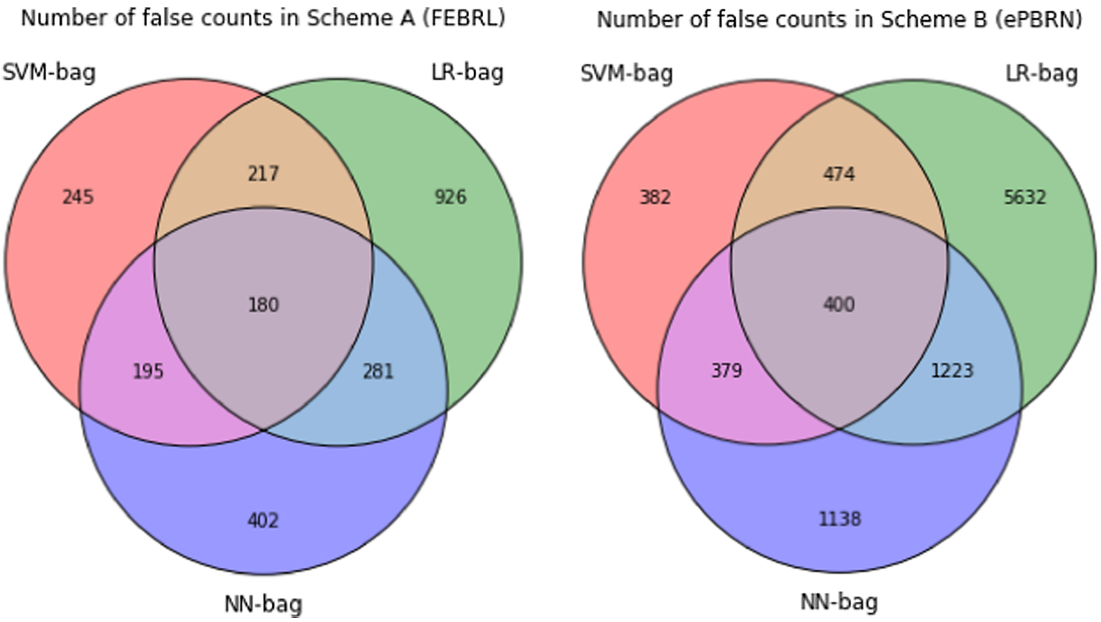
	Scheme A				Scheme B			
	pr(%)	re(%)	fs(%)	fc	pr(%)	re(%)	fs(%)	fc
SVM	94.85	99.73	97.23	278	85.84	96.08	90.67	514
SVM-bag	95.46	99.73	97.55	245	99.25	86.84	92.63	359
NN	92.80	99.59	96.08	398	69.08	96.46	80.51	1214
NN-bag	92.75	99.57	96.04	402	70.56	96.46	81.50	1138
LR	84.46	99.69	91.44	913	31.61	97.80	47.78	5556
LR-bag	84.27	99.69	91.33	926	31.30	97.69	47.41	5632
Stack & Bag	96.97	99.43	98.18	180	99.28	85.23	91.72	400

Table 6 presents the results of all base learners and ensemble methods. We will explain each scheme's results of each base learner, and subsequently discuss the influence of bagging and stacking on each scheme. The metrics used in this study are precision (pr), recall (re), F-score (fs), and false counts (fc). The results indicate that our meta-ensemble, compared to base learner, performed significantly better in one dataset (FEBRL), and equivalent to the best performing base learner (SVM-bag) in the other dataset (ePBRN). Specifically, comparing the average performance of three base learners with the stacking ensemble results, we observed a boost in precision from 90.70% (as the average of 94.85% of SVM, 92.80% of NN, and 84.46% of LR) to 94.85% on FEBRL dataset, and a boost from 62.17% (as the average of 85.84% of SVM, 69.08% of NN, and 31.61% of LR) to 99.28% in ePBRN dataset. Please refer to Appendix B for additional details on metrics used.

4. Discussion

The three-criteria blocking technique detected 97.88% of matched pairs in scheme A and 95.28% in scheme B. The ratio of extracted candidates to the maximum number of candidates were 0.74% for scheme A and 0.48% for scheme B, implying that we have significantly reduced the number of test feature vectors for both schemes by over 99%. However, when individual criterion was, many candidate pairs were missed. The combination of all criteria ensures a high performance but with a tradeoff in the nc.

In scheme A, SVM and NN obtained high scores in terms of precision, recall, and F-score (above 90%), while LR underperformed. Bagging performed well for SVM but slightly better in NN and worse in LR. The most important observation in scheme A was that all base learners performed almost equivalently. Consequently, the final stacking ensemble performed significantly better than any given single model. The precision and F-score of the best bagging model (SVM-bag) improved from 95.46% to 96.97%, and from 97.55% to 98.18%, respectively with our meta-ensemble. The number of false counts (FP + FN) decreased from 245 to 180 cases. For scheme B, each base learner performed differently. The F-score for SVM, NN, and LR were 90.67%, 80.51%, and 47.78% respectively. LR severely suffered from underfitting due to its probabilistic approach which takes all training samples into consideration. Although rbf-kernel SVM was opted as the result of our hyperparameter tuning, it is noteworthy that LR was outperformed by another linear classifier, namely linear-kernel SVM in scheme B with a F-score of 80.05%. Generally, SVM performed the best in terms of all metrics. We can attribute this to the synthetic training data we generated that covered very samples that require sensitive SVM boundaries. For instance, a pair of two true linked records with different dates of birth was assigned as negative by LR and NN but positive by SVM. LR, with probabilistic boundary, could not capture these cases while the NN assigned large weights. SVM was able to capture as we employed the RBF kernel function that increased the dimension space [27]. Finally, because of the performance gap between the base learning models was large, the stacking ensemble deteriorated slightly. The F-score decreased from 92.63% to 91.72% when compared to the best performing SVM-bag model. Fig. 3 depicts the false counts of different ensemble combinations. In scheme A, every combination resulted in fewer false counts. Meanwhile, in scheme B, only the combination of SVM-bag and NN-bag yielded better performance, and the LR-bag decreased the


Number of false counts in Scheme A (FEBRL)

Number of false counts in Scheme B (ePBRN)

Fig. 3. Number of false counts in Scheme A and Scheme B.

performance of the other members in every combination it participated.

One major advantage of our proposed ensemble method is its tolerance against overfitting as well as its capability to significantly boost the overall performance. When there is one base learner overfitting on the training data, the other base learners in the ensemble scheme serve as a means of regularization to cancel out the overfitting issue. When all base learners underfit the training data, the ensemble can combine their out-of-fold predictions to yield a better performance. However, this ensemble suffers from various limitations. A large training data is required to better exploit the ensemble benefits. Also, training and validating on out-of-fold training samples need to be carefully implemented to avoid data leak, which is undesirable for the stacking algorithm [21].

We have presented summary of various state-of-the art record linkage methods in comparison with our ensemble in Appendix C. It is vital to note that the performance measures presented in this appendix are not directly comparable. Each method has been evaluated in a different setting. Theoretically the ensemble based methods are supposed to work better than a single base learner and this has been evident in recent machine learning competitions [32,33,34].

5. Conclusion

We proposed and tested ensemble-based method to link records in two datasets, created using record linkage synthetic data generators. Our meta-ensemble is a combination of bagging and stacking ensembles. The training sets were divided into different continuous folds, where each fold was validated by the model trained on all other folds. The resulting model was then evaluated on the test sets using SVM, NN, LR and ensembles. Critical hyperparameters of base learners were selected using grid search. The results indicate that the final meta-ensemble compared to the base learners performed significantly better on one dataset (FEBRL), and equivalent to the best performing base learner (SVM-bag) on the other dataset (ePBRN). These results highlight that the meta-ensemble performance is highly dependent on the base learners. We believe our meta-ensemble can be further improved by incorporating more advanced features. Feature extraction is important in improving the base learners' performance, especially for NN and LR. In future, the diversification of the base learner collection will be examined to assess the effectiveness of meta-ensemble.

Acknowledgments

This study was funded by the UNSW Research Infrastructure Scheme. It was conducted as part of the UNSW Electronic Practice Based Research Network (ePBRN) research program. The ePBRN is supported in part by UNSW Medicine (School of Public Health & Community Medicine and Centre for Primary Health Care & Equity), Ingham Institute for Applied Medical Research, and South West Sydney Local Health District. The content of this publication is solely the responsibility of the authors and does not necessarily reflect the official views of the funding bodies.

Appendix A

The electronic practice-based research network (ePBRN) dataset

The University of New South Wales (UNSW) electronic Practice Based Research Network (ePBRN) has been extracting clinical and administrative data from electronic health records (EHRs) for research and quality improvement purposes. This provides real-world observational data to improve the understanding and knowledge regarding diseases, diagnoses, treatments, health practice, and health outcomes. However, data from single EHRs only provide part of the patient's history and journey through the healthcare system. Data must be collected from multiple EHRs from both primary and secondary care services to provide a more comprehensive picture of the patient's journey through the health system. While the secondary use of linked health data for research exhibits great potential, significant challenges exist such as the quality of the routinely collected data. The data quality can affect the accuracy of the data linkage techniques as well. The following tables present a quick overview of the ePBRN data repository summary level statistics based on data extracted in Nov 2017.

Age groups	All patients (n = 212144)	
	Female Count (%)	Male Count (%)
0–5 yrs	5182 (4.7)	5844 (5.8)
6–19 yrs	18,657 (17.0)	20,192 (19.9)
20–44 yrs	43,294 (39.4)	37,275 (36.8)
45–64 yrs	27,611 (25.1)	24,852 (24.5)
65–74 yrs	7903 (7.2)	7280 (7.2)
75+ yrs	7202 (6.6)	5845 (5.8)
Total patients	109,948 (51.8)	101,403 (47.8)

Appendix B

Evaluation metrics

Pair completeness is the ratio of matched pairs after blocking to the total matched pairs (Eq. (15)). Reduction ratio is the ratio of number of candidate pairs after blocking compared to total possible number of candidate pairs (Eq. (16)).

$$pc = \frac{Numberofmatchedpairsafterblocking}{Totaltruematchedpairs} \tag{15}$$

$$rr = \frac{Numberofpairsafterblocking}{Numberoftotalpossiblepairscreated}. \tag{16}$$

The evaluation metrics used in this study are precision (pr), recall (re), F-score (fs), and false counts (fc), which can be computed as follows, respectively:

$$\begin{aligned} pr &= TP/(TP+FP)\times 100(\%),\\ re &= TP/(TP+FN)\times 100(\%),\\ fs &= 2\times pr\times re/(pr+re)\times 100(\%),\\ fc &= FP+FN, \end{aligned} \qquad (17)$$

where TP (true positive), FP (false positive), FN (false negative), and TN (true negative) are retrieved from the confusion matrix.

		True label	
		Match	Non-match
Predicted	Match	*TP*	*FP*
	Non-match	*FN*	*TN*

Appendix C

Summary of performance of related studies. It is important to note that each method presented in this table are evaluated under different settings and not directly comparable.

Method	Year	Algorithm	Dataset	*pr*	*re*	*fs*
Conrad et al. [1]	2016	C4.5	proprietary data of customers and their political affiliations	0.699	0.696	0.692
		C4.5	proprietary data of customers and their political affiliations	0.713	0.713	0.709
Paixao et al. [35]	2017	Custom probabilistic algorithm	Dengue and still birth datasets from Brazil	0.684	0.691	0.687
		RecLinkIII	Dengue and still birth datasets from Brazil	0.145	0.597	0.233
Pita et al. [36]	2018	The 114 Million Cohort	AtyImo	0.959	0.943	0.950
		The 114 Million Cohort	AtyImo	0.977	0.976	0.976
		The 114 Million Cohort	AtyImo	0.935	0.866	0.899
Our method	2019	Stacking and bagging	FEBRL	0.969	0.994	0.981
		Stacking and bagging	ePBRN	0.992	0.852	0.917

References

[1] C. Conrad, N. Ali, V. Keelj, Q. Gao, ELM: an extended logic matching method on record linkage analysis of disparate databases for profiling data mining, in: 2016 IEEE 18th Conference on Business Informatics (CBI), vol. 01, 2016, p. 1–6.

[2] Y. Jeon, J. Yoo, J. Lee, S. Yoon, NC-link: a new linkage method for efficient hierarchical clustering of large-scale data, IEEE Access 5 (2017) 5594–5608, https://doi.org/10.1109/ACCESS.2017.2690987.

[3] H.-S. Kim, D. Lee, Harra: Fast iterative hashed record linkage for large-scale data collections, in: Proc. 13th Int. Conf. Extending Database Technol., 2010, pp. 525–536.

[4] T. Sagi, A. Gal, O. Barkol, R. Bergman, A. Avram, Multi-source uncertain entity resolution at yad vashem: Transforming holocaust victim reports into people, in: Proc. Int. Conf. Manage. Data, 2016, pp. 807–819.

[5] S. Rendle, L. Schmidt-Thieme, Scaling record linkage to non-uniform distributed class sizes, in: T. Washio, E. Suzuki, K.M. Ting, A. Inokuchi (Eds.), Advances in Knowledge Discovery and Data Mining, Springer, Berlin Heidelberg, 2008, pp. 308–319.

[6] E. Begoli, T. Dunning, C. Frasure, Real-time discovery services over large, heterogeneous and complex healthcare datasets using schema-less, column-oriented methods, in: 2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService), 2016, pp. 257–264.

[7] A.-A. Mamun, T. Mi, R. Aseltine, S. Rajasekaran, Efficient sequential and parallel algorithms for record linkage, J. Am. Med. Informat. Assoc. 21 (2) (2014) 252–262.

[8] P. Christen, Febrl - An open source data cleaning, deduplication and record linkage system with a graphical user interface (Demonstration Session, ACM International Conference on Knowledge Discovery and Data Mining) SIGKDD'08, 2008, pp. 1065–1068.

[9] P. Christen, T. Churches, M. Hegland, Febrl - a parallel open source data linkage system, in: Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science), vol. 3056, 2004, pp. 638–647.

[10] P. Jurczyk, J. Lu, L. Xiong, D. Cragan, J.A. Correa, FRIL: A tool for comparative record linkage, 2008, 2008: pp. 440–444.

[11] R.C. Steorts, S.L. Ventura, M. Sadinle, S.E. Fienberg, A comparison of blocking methods for record linkage, in: J. Domingo-Ferrer (Ed.), Privacy in Statistical Databases, Springer International Publishing, Cham, 2014, pp. 253–268.

[12] R. Baxter, P. Christen, For Epidemiology C. A Comparison of Fast Blocking Methods for Record Linkage, 2003.

[13] T. Churches, P. Christen, K. Lim, J. Zhu, Preparation of name and address data for record linkage using hidden Markov models, BMC Med. Inf. Decis. Making 2 (2002) 9.

[14] M.F. Lima-Costa, L.C. Rodrigues, M.L. Barreto, M. Gouveia, B.L. Horta, J. Mambrini, et al., Genomic ancestry and ethnoracial self-classification based on 5,871 community-dwelling Brazilians (The Epigen Initiative), Nat. Sci. Reports 5 (2015) 9812, https://doi.org/10.1038/srep09812.

[15] K.V. Freitas De Andrade, J. Silva Nery, G. Oliveira Penna, L. Penna, M. Barreto, Pereira S. Martins, Effect of Brazil's Conditional Cash Transfer Programme on the new case detection rate of leprosy in children under 15 years old, PLoS Negl.Trop. Dis. 8 (11) (2014) 1–7.

[16] V. Efthymiou, G. Papadakis, G. Papastefanatos, K. Stefanidis, T. Palpanas, Parallel meta-blocking for scaling entity resolution over big heterogeneous data, Inform. Syst. 65 (2017) 137–157, https://doi.org/10.1016/j.is.2016.12.001.

[17] R. Schnell, T. Bachteler, J. Reiher, Privacy-preserving record linkage using Bloom filters, BMC Med. Inf. Decis. Making 9 (1) (2009) 41, https://doi.org/10.1186/1472-6947-9-41.

[18] P. Christen, Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection (Data-Centric Systems and Applications), 2012th ed., Springer, 2012.

[19] J. Turk, M. Stephens, Jellyfish: a python library for doing approximate and phonetic matching of strings. Availability: https://jellyfish.readthedocs.io/en/latest/.

[20] Z.H. Zhou, Ensemble Methods: Foundations and Algorithms, 1st ed., Chapman & Hall/CRC, 2012.

[21] L. Breiman, Stacked regressions, Mach. Learn. 24 (1) (1996) 49–64, https://doi.org/10.1007/BF00117832.

[22] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140, https://doi.org/10.1023/A:1018054314350.

[23] Y. Grandvalet, Bagging equalizes influence, Mach. Learn. 55 (3) (2004) 251–270, https://doi.org/10.1023/B:MACH.0000027783.34431.42.

[24] Astrid Schneider, Gerhard Hommel, Maria Blettner, Linear Regression Analysis: Part 14 of a Series on Evaluation of Scientific Publications, Deutsches Ärzteblatt International 107 (44) (2010) 776–782 PMC. Web. 15 Aug. 2018.

[25] Siaw-Teng Liaw, Jane Taggart, Yu Hairong, Simon de Lusignan, Data extraction from electronic health records - existing tools may be unreliable and potentially unsafe, Aust. Family Phys. 42 (11) (2013) 820–823 ISSN: 0300-8495. [cited 05 Sep 18].

[26] Siaw-Teng Liaw, Gawaine Powell-Davies, Christopher Pearce, Helena Britt, Lisa McGlynn, Mark Fort Harris, Optimising the use of observational electronic health record data: Current issues, evolving opportunities, strategies and scope for collaboration [online]. Aust. Family Phys. 45(3), Mar 2016: 153-156.

Availability: < https://search.informit.com.au/documentSummary;dn=926994450045520;res=IELHEA > ISSN: 0300-8495. [cited 05 Sep 18].

[27] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297, https://doi.org/10.1007/BF00994018.

[28] S.H. Walker, D.B. Duncan, Estimation of the Probability of an Event as a Function of Several Independent Variables, Biometrika 54 (1/2) (1967) 167–179.

[29] G. Cybenko, Approximation by Superpositions of a Sigmoidal Function, 1989 August 15, 2018.

[30] V. Nair, G.E. Hinton, Rectified Linear Units Improve Restricted Boltzmann Machines, in: Proceedings of the 27th International Conference on International Conference on Machine Learning. ICML'10. Omnipress, USA, 2010, pp. 807–814.

[31] P. Christen, K. Goiser, Quality and Complexity Measures for Data Linkage and Deduplication, Springer, Berlin, Germany, 2007, pp. 127–151.

[32] Kyle Boone, PLASSITICC astronomical classsification Competition 1st place solution, on Kaggle, 2019. Available at https://www.kaggle.com/c/vsb-power-line-fault-detection/discussion/87038#latest-504346.

[33] Xiao Zhuwang, AVITO demand prediction Competition 1st place solution, on Kaggle, 2018. Available at https://www.kaggle.com/c/avito-demand-prediction/discussion/59880#latest-450523.

[34] Gilberto Titericz, Santander value prediction Competition 1st place solution, on Kaggle, 2019. Available at https://www.kaggle.com/c/santander-value-prediction-challenge/discussion/63907#latest-380059.

[35] E.S. Paixao, K. Harron, K. Andrade, M.G. Teixeira, R.L. Fiaccone, M.C.N. Costa, et al., Evaluation of record linkage of two large administrative databases in a middle income country: stillbirths and notifications of dengue during pregnancy in Brazil, BMC Med. Inform. Decis. Mak. 17 (2017) 108.

[36] R. Dantas Pita, C. Pinto, S. Sena, R. Fiaccone, L. Amorim, S. Reis, M. Barreto, S. Denaxas, M. Barreto, On the accuracy and scalability of probabilistic data linkage over the Brazilian 114 million cohort, IEEE J. Biomed. Health. Inf. 22 (2) (2018) 346–353.