



Attended robotic process automation of prescriptions' digitization

Kabir Soeny^{*}, Gaurav Pandey, Utkarsh Gupta, Adarsh Trivedi, Mehul Gupta, Gaurav Agarwal

1mg Technologies Private Limited, Gurugram, India



ARTICLE INFO

Keywords:

Artificial intelligence
Image processing
Applied machine learning
Intelligent systems
Healthcare technology

ABSTRACT

Document digitization has important industrial applications and automation of this process can unlock significant business value, especially when documents are in the form of image files that are not directly machine readable. A medical prescription contains information about a patient's health status and is an important tool for prevention of drug abuse and patient safety. In our organization, in the interests of patient safety and to minimize dispensation errors, the customer is required to upload a picture of the prescription with her order if it contains any prescription drug. Once this prescription is digitized, the system decides the eligibility for dispensation of the ordered medicines based on the information extracted from the prescription. Since we process thousands of medicine orders everyday for which a large number of man-hours are required for processing the uploaded prescriptions, a key opportunity was identified in this process to make it more efficient and faster. In this paper, we describe how artificial intelligence and ingenious solutions have enabled us to automate different elements of the screening process and present metrics for evaluating the performance of the automated process. Our paper demonstrates that attended robotic process automation reduces the human effort to verify a prescription and facilitates a superior employee experience. We also underpin the importance of prescription data to the healthcare community at large and advocate that such automations further the cause of healthcare research.

1. Introduction

A prescription is a doctor's order which stipulates administration of drugs in the specified amount, duration and frequency, and contains details of the patient such as name, age and gender, and also the details of the doctor who writes the prescription. It is a common clinical practice to also write about the patient reported complaints and the diagnosis that the doctor arrived at after clinical examination. The symbol 'Rx' denotes a prescription and is often inscribed in the header of a prescription. It stands for the Latin word *recipe* which means 'to take'.

A prescription ensures that the information about the prescribed drugs is passed on accurately to a pharmacist who then ensures that the drugs are dispensed without any errors and in accordance with the laws and regulations governing the dispensation of drugs. While a prescription is a time-tested device to prevent drug abuse and wrong dispensation, it is not without shortcomings. Misinterpretation of the prescribed drugs, for reasons that include illegible handwriting and lack of experience of the pharmacist, result in erroneous dispensation of medications which often lead to serious consequences for the patient and the healthcare system, (Aronson,

* Corresponding author.

E-mail address: kabir.soeny@1mg.com (K. Soeny).

2009; Caban et al., 2012). E-prescriptions, which consist of electronic generation and transmission of the prescribed medicines' data directly to the pharmacists' computers, have been suggested as a solution to reduce the paperwork and medication errors. However, such systems are also not without drawbacks with concerns being raised about costs, patient privacy and legal issues, (Porterfield et al., 2014). As such, paper prescriptions continue to be the mainstay of the medical systems in most parts of the world.

Our company, 1mg Technologies, 2020, is a digital healthcare platform, which offers services such as electronic ordering of medicines, healthcare products and diagnostic tests, and teleconsultations. Any order which contains at least one prescription medicine, has to be accompanied, as per government laws and regulations, with a *valid prescription* - defined as one which contains details of the patient such as name, age and date of prescription, and the name and signature of a registered doctor on official letterhead. Furthermore, the medicines that are ordered must be present in the accompanying prescription and the duration and strength of the medicines should be consistent with the information mentioned in the prescription.

This requirement, while ensuring compliance and patient safety, implies that we need to operate a process and invest trained and qualified manpower to peruse every uploaded image of the prescription for its validity. Given that we receive more than ten million orders annually, we need to invest thousands of man-hours in the task of searching for the relevant information in the every prescription that is uploaded on our platform and entering the scoured information in a proprietary tool for auditing purpose. The tool then feeds this information to our data warehouses which securely store this information for every prescription against the corresponding user identification key.

While this process serves our objectives, we realized that several aspects of this process can be assisted by a machine to reduce the human effort and processing time that is expended per prescription. While traditional automation is general in nature that seeks to automate any task or step in a process, robotic process automation (RPA) pertains to artificial intelligence (AI) powered computer software or 'bots' to emulate the actions of a human agent in interaction with the computer, (Asquith & Horsman, 2019). RPA solutions work at the user interface (UI) level and are generally easier to integrate in the existing systems as compared to traditional automation technologies. RPA solutions reduce the burden of repetitive and simple tasks on employees and are recognized for achievement of high return on investment, (Van Der Aalst et al., 2018). Furthermore, research shows that Human-Robot Interaction and robot companionship lead to positive results on the human psychological state, (Madakam et al., 2019).

A further distinction is drawn between attended RPA and unattended RPA. While attended RPA bots behave like virtual assistants by helping the employees with their repetitive tasks, unattended RPA bots execute end-to-end business processes independently, (Anywhere, 2020). Also, while the attended RPA bots' actions can be overridden by the collaborating human, unattended RPA bots, because of lack of human involvement, go exactly as per the pre-defined rules. Therefore, while both attended and unattended RPAs lead to increased productivity, attended RPA, generally, is better equipped to handle unexpected scenarios because of the accompanying human intelligence.

The use of computers for converting documents to a machine readable format is not a new problem. In fact, the first optical character reader (OCR) that could convert printed messages into machine language was invented in 1951, (Shepard, 1951). Today, with the availability of high quality OCR engines and the drive for digital transformation in most industries, document digitization, including digitization of medical records, is an active area of research. For example (Fajardo et al., 2019), describe the use of deep learning-based classification models for detection of medicines' names in handwritten prescriptions. The authors test their methodology on a target set of twelve medicines. Most of the research in the area of prescriptions' digitization is focused on detection of medicines' names only and the target set is generally a small number of commonly prescribed medicines. That is, often it is a classification problem and may not perform adequately when applied on a large set of medicine stock keeping units (SKUs). Furthermore, the extraction of other potentially valuable information such as name of the patient, age, prescribing doctor's name and date of appointment have not been explored in detail.

In this paper, we describe how we integrated data science models seamlessly within a process that requires heavy human interaction. We demonstrate that the attended RPA enabled our process of prescription validation and digitization to run decidedly faster and with reduced human effort. We also discuss the benefits that accrue to the business as well as to the medical research community from the data that this process generate.

The novelties of our framework, which is built on top of the state-of-the-art Google Vision OCR, are as follows. Firstly, we extract not just the names of the medicines, but also other information contained in the prescription, leading to the creation of a digital health record for the patient. Secondly, our methodology for detecting medicines' names is not based on a classification algorithm and, therefore, is not limited to a pre-defined target set of medicines' names. The methodology works for about 95,000 medicine SKUs that we have in our system. Thirdly, since our framework is assistive, the human-machine collaboration ensures that the final data that get stored in the database have high accuracy and quality. Fourthly, we recognize the inherent differences between handwritten and printed prescriptions and develop customized methods for the two categories of prescriptions, along with a mechanism for auto-detection of the prescription type. Finally, we demonstrate, over a large corpus of more than 115,000 prescriptions, the effectiveness of our methodology.

The paper is organized as follows: Section 2 explains the current process of prescription digitization and validation. Section 3 presents the assisted RPA system which consists of a suite of models for prediction of various elements of a prescription. Section 4 presents the framework to evaluate the performance of the assisted RPA process. In Section 5, we present the results from the experiment which prove the effectiveness of the assisted RPA system. Finally, in Section 6, we discuss a few applications of the digitized data generated from prescriptions.

2. The current process

To order a SKU which is marked with 'Rx', the system requires that the patient uploads a *valid* prescription. Every order that contains at least one Rx labelled SKU passes through a rigorous screening process in which the prescription is scrutinized for the

Medicines Ordered (1)
Delivery Location Karnataka

Sporanox Capsule Generated by System

Itraconazole (100mg) Strength

Please validate the prescription before checking for order validity.

Reject **Accept**

(a) S1 section of the screening tool

Medicines Ordered (1) Screening-1 status Accepted
Delivery Location Delhi

Nexito 5 Tablet

Escitalopram Oxalate (5mg) Strength Chronic

Dosage: 1 tablet One Time A Day for 730 days

Reject **Accept**

(b) S2 section of the screening tool

Fig. 1. Snapshots of (a) S1 section where patient's name, age, visit date are entered and the presence of the ordered SKUs is confirmed and (b) S2 section where name of the doctor and the dosage, frequency and the duration of the drug is entered. Personally identifiable information (PII) of the customers has been masked in the images.

presence of all of the following data points:

1. Name of the patient,
2. Age of the patient,
3. Visit date (prescription date),
4. Name of the doctor, and
5. Presence of the ordered medicines in the prescription.

In the current process, the employees who are tasked with screening these prescriptions (henceforth referred to as the *screening staff* or the *screeners*), have to search for these values in the uploaded image of the prescription and then enter these values manually in a proprietary *screening tool*.

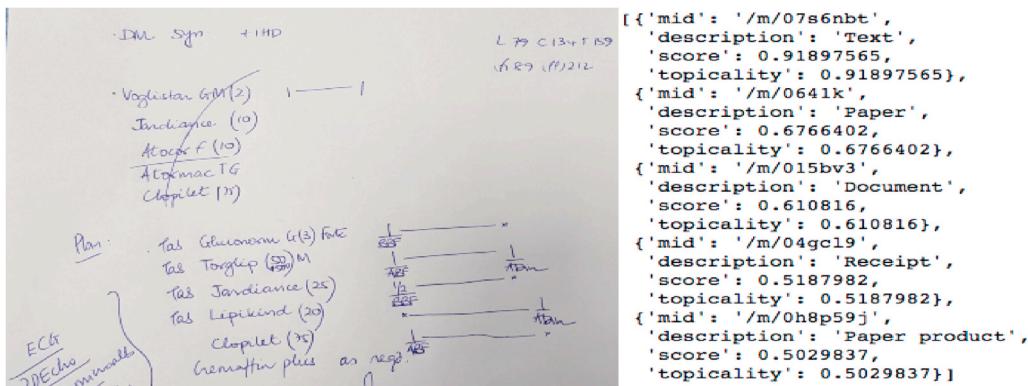
The screening tool has two sections: (a) In S1, the screening staff have to enter the patient's name, age, the date of visit and also confirm the presence of the ordered SKUs in the prescription's image. In case any field is missing, or if each ordered medicine is not present on the uploaded copies of the prescriptions, the order is rejected.

Post successful validation in the S1 section, the prescription moves into the second section, that is, S2. In S2, the name of the doctor and the dosage, frequency and the duration of the drug is entered by the screening staff. For example, if the recommended strength of a medicine in the prescription is 50 mg and the customer orders for 75 mg, such an order will be rejected from further processing. Upon completion of these fields, the screening of the order is said to be complete. In case a medicine was incorrectly passed in S1 to be present in the prescription, S2 gives an opportunity for the mistake to be corrected, and for the order to be on hold pending a valid prescription.

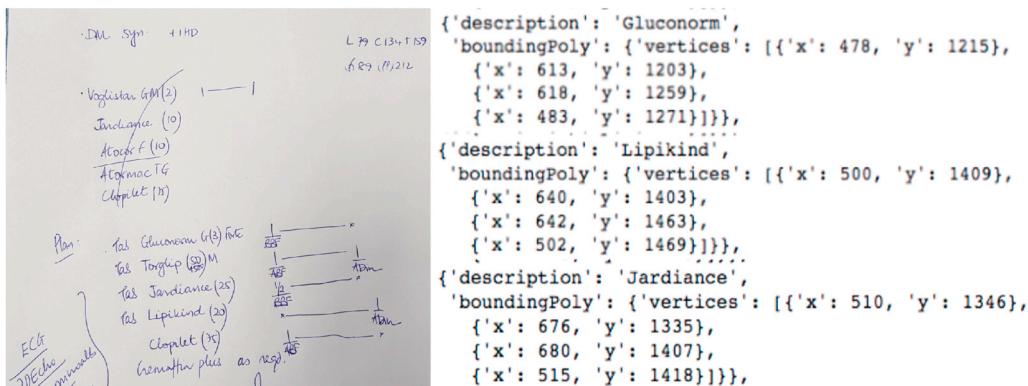
The objective of having two separate screening sections is that once S1 is completed, the order flows ahead for further processing while S2 is getting processed in parallel which saves overall processing time. In case any mistakes are identified in S2, orders can be clawed back from the processing queue and kept in a waiting queue pending a valid prescription.

Fig. 1 presents snapshots of the S1 and S2 sections of the screening tool.

In the current process, once the drug has been identified, the dosage, frequency and the duration of the drug are auto-filled in the



(a) Partial output from the LabelAnnotation key



(b) Partial output from the FullTextAnnotation key

Fig. 2. Partial image of a prescription uploaded at our platform and the output returned from the Google Vision API.

tool based on known medical knowledge. However, other elements of the prescription have to be entered manually in the tool.

Screeners' efforts are towards two activities - Firstly to locate the relevant fields in the prescription image and secondly, to type these values in the relevant text boxes (except for medicines - a missing medicine is simply indicated by clicking the 'X' next to it). In the next section, we present an artificial intelligence (AI) based framework which reduces both kinds of efforts thereby reducing the average order processing time.

3. AI based process automation framework

In this section, we present the suite of models that are part of the framework that we have developed to assist the current digitization process.

3.1. Google Vision API

Google Cloud's Vision API offers powerful pre-trained machine learning models through REST APIs, ([Google, 2020](#)). It assigns labels to images and is able to classify them into millions of predefined categories. It is capable of detecting objects and reading printed and handwritten text.

Our framework is built on top of the output returned from the Vision API. Every prescription uploaded on our platform is sent to the Vision API and the response is a JavaScript Object Notation (JSON) file which, in our context, consists of the following parts:

- LabelAnnotation

This level has information about the entities or attributes found in the image. In our case, for images of prescriptions, some common entities can be paper, document and text.

LabelAnnotation has 4 major parts:

(a) 'mid': A unique machine-generated identifier, (b) 'description': The label description, (c) 'score': A confidence score between 0 and 1, and (d) 'topicality': The relevancy of the label to the image. It measures how important a label is to the overall context of a page.

[Fig. 2a](#) presents a part of a prescription uploaded on our platform and the LabelAnnotation part of the JSON output from the Vision API.

- FullTextAnnotation

The second level, FullTextAnnotation, is a structured hierarchical response for the UTF-8 text extracted from the image. The hierarchy is organized in the following order:

Pages > Blocks > Paragraphs > Words > Symbols. We briefly describe these below:

Page is a collection of blocks and meta-information about the page: image size and resolutions. Block represents one logical element - area covered by text or a picture - of the page. A page can have multiple blocks. The text blocks contain the main information needed to extract the text: four (x,y) co-ordinate pairs starting from top left of the block, the associated confidence scores and paragraphs.

Paragraph is a structural unit of text representing an ordered sequence of words. By default, words are considered to be separated by word breaks. A paragraph provides the following information: four (x,y) co-ordinate pairs starting from top left of the paragraph, the associated confidence scores and words.

Word is the smallest unit of text. It is represented as an array of Symbols. A word provides the following information: four (x,y) co-ordinate pairs starting from top left of the word, the associated confidence scores and symbols.

Symbol represents a character or a punctuation mark and is at the lowest level in the hierarchy. A symbol provides the following information: language for that symbol or letter, four (x,y) co-ordinate pairs starting from top left of the symbol or letter, the associated confidence scores and text of the symbols.

[Fig. 2b](#) presents some of the output from FullTextAnnotation for part of a prescription's image uploaded on our platform.

It must be noted that any sort of text in FullTextAnnotation is present at Symbol level only. Hence, to fetch the text corresponding to any entity (such as Blocks, Paragraph or Words) it has to be extracted from a Symbol only.

The co-ordinate system according to which co-ordinates for various entities are provided assumes (0, 0) at the top left corner of the image irrespective of its orientation. The four (x,y) co-ordinate pairs in every entity start from the top left corner of the entity (and not the image) and the sequence for the rest of the 3 coordinate pairs are given in a clock-wise order. Hence, same image with different orientations (rotated at different angles) will provide different co-ordinates for the same entities.

3.2. Suite of data science models

We now define the suite of models that we have developed and deployed (in Python language environment) for prediction of various elements of a prescription in the screening process. All our models and procedures rely on the JSON output returned from the Google Vision API. That is, any data mining or extraction of features is carried out from this output only. We describe these methods below.

3.2.1. Prediction of Doctor's name

The steps to predict the name of the doctor in a prescription are as follows: [enumerate]

1. From the JSON output obtained from the Vision API, we extract the FullTextAnnotation key.
2. From the key obtained in Step 1, the following two methods are used in parallel to extract possible names of the doctor:
 - (a) Regular expression method: A regular expression is a sequence of characters that define a search pattern. A regular expression software processes regular expressions, trying to match the pattern to the given string, (Thompson, 1968). We convert the words found in the first step to lower case and select strings starting with tokens such as 'dr', 'dr.' and 'doctor' using the regular expression library (*re*) in python.
 - (b) Named Entity Recognition (NER) method: NER is the task of identifying named entities such as person, location, organization in text. NER systems are often used as the first step in information retrieval and topic modeling, (Yadav & Bethard, 2018). Python's *spaCy* library is used to implement NER and obtain a list of names of persons.
3. The strings obtained from the above two methods are stored in one list by giving priority to the regular expression method in 2(a). The first string in this list is shown as a suggestion for the doctor's name in the screening tool, as can be seen in Fig. 1b.

3.2.2. Prediction of patient name

For determination of the patient's name on a prescription, we follow these steps:

1. From the JSON output obtained from the Vision API, we extract the FullTextAnnotation key and pull the historical patient names, if any, associated with the patient ID for which the prescription has been uploaded.
2. We use Python's *spaCy* library to perform point-of-speech (POS) tagging, (Neunerdt et al., 2013), on the FullTextAnnotation key and extract the tokens with a 'proper noun' tag.
3. Using regular expression method, select strings containing expressions such as 'patient name', 'patient's name', 'patient', 'name'.
4. The strings common to both Steps 2 and 3, that is, which have tags of proper noun and also contain the expressions contained in Step 3, are taken forward for further processing. From this list, the names that are predicted to be of the doctor (using the previous method) are expunged so that these names are not considered as suggestions for the name of the patient.
5. In the list obtained from the previous step, we rank a string higher if it contains words such as 'Mr.', 'Ms', etc. After ranking, these salutations are removed from the strings in order to extract the clean string containing only potential names of the patient.
6. This final processed list obtained in the previous step is matched, using fuzzy logic, with the historical names found in Step 1, (Ukkonen, 1985). Python's *FuzzyWuzzy* library is used for this matching which returns a 'fuzzy score', ranging from 0 to 100, for a given pair of strings. A higher score is associated implies greater similarity between the strings.
7. Based on the fuzzy score, one of the two steps is followed:
 - (a) If the fuzzy score is greater than 60 for any historical patient name, that name is the output from the data science model and gets pre-filled in the patient name text box.
 - (b) If the fuzzy score is less than or equal to 60 or if there are no historical names associated with the patient ID, the first element of the list obtained in Step 5 is the output from the data science model and gets pre-filled in the patient name text-box.

The output from the above algorithm is then pushed to the front-end and is pre-filled in the text-box for the patient's name, as shown in Fig. 1a (the name is concealed as it is PII).

3.2.3. Prediction of Patient's age

1. From the JSON output obtained from the Vision API, we extract the FullTextAnnotation key.
2. We then use regular expression method to fetch one or two digits numbers in close proximity of the following expressions (in the listed order of precedence):
 - (a) Expressions such as 'years', 'year', 'yrs', 'y'.
 - (b) Expression similar to 'age'.
 - (c) Expression such as 'patient' and 'patient name'.
 - (d) Expression such as 'male', 'm', 'f', 'female'.

This precedence list was determined through several experiments.

3. A ranked list, according to the above level of precedence is then prepared and the first element of that list is the suggestion, which gets pre-filled in the screening tool as shown in Fig. 1a. In case none of the above regular expression searches return any suggestion, the age input box is left blank.

3.2.4. Prediction of visit date

1. From the JSON output obtained from the Vision API, we extract the FullTextAnnotation key.
2. We then use Python's *date-extractor* library to fetch all dates contained in the above key.
3. The list of dates is then sorted in order of their proximity to the order placement date.

4. The date that ranks highest in this list is then pre-filled in the Visit Date input box, as can be seen in Fig. 1a. It remains empty in case no dates are detected from the above approach.

3.2.5. Detection of ordered medicines in the prescription

The problem of detection of the ordered medicines in the uploaded image of the prescription, is much more complex than the other elements of the prescription that have been discussed so far. Especially for handwritten prescriptions, since the text detected by the vision API in the prescription can deviate significantly from what the actual text is, the challenge of detecting the presence of a given medicine in the prescription is hard.

The data science intervention in the case of medicines is the display of “verified by system” tag in the screening tool against the medicines whose presence in the prescription had been detected by the models. For the medicines not found by the models in the prescription, there is no such tag. Therefore, this is a binary classification problem, with the two classes being *Medicine Found* (presented as “verified by system” tag to the screeners) and *Medicine Not Found* (no tag) in the S1 section of the screening tool. The process is designed in a manner that every misclassification gets remedied - a medicine wrongly found to be present in S1 will be caught in the S2 section and the medicines that are wrongly found to be absent, anyway necessitate supervision by a screener. Both these actions would increase the processing times and would tend to nullify the savings induced by these models.

In the context of a binary classification problem, it is pertinent to have three metrics for measuring a model’s performance. *Accuracy*: the sum of True Positives and True Negatives divided by the total number of observations. *Precision*: ratio of number of True Positives to the number of observations predicted as positive. *Recall*: ratio of the number of True Positives to the sum of True Positives and False Negatives.

In order to solve the above problem of accurately predicting the status of an ordered medicine in the prescription, we had to develop two accompanying binary classification models. Before we describe the solution to the overall problem, we describe these two models below.

3.2.5.1. Prescription type classifier. The difference in the legibilities of a handwritten and a printed prescription is often apparent through naked eyes. This difference gets reflected in the quality of the output received from the Vision API as well. While the average confidence score (as described in Section 3.1) per word for a printed prescription is about 0.95, the same metric for a handwritten prescription is only 0.70. This difference necessitates development of separate approaches for handling the two prescription types. As we will describe later in the paper, while fuzzy matching works well for a printed prescription because of the high confidence score of the words, the same technique performs poorly for handwritten prescriptions as the words received from the Vision API are significantly different from the actual words (that is, the words have low confidence scores). Therefore, we had to develop a separate approach for handwritten prescriptions, as described later in the paper.

However, to implement separate approaches, a prescription first needs to be classified as printed or handwritten. Assigning the responsibility of this classification to the screeners would have added to their workloads. Therefore, we developed a predictive model which uses the features extracted from FullTextAnnotation and LabelAnnotation to predict the type of the prescription, which is then used to select the appropriate strategy.

Towards this end, we developed several models and benchmarked their performances through the above three metrics and the AUC.

A five-layers dense neural network model, trained using the Keras library in Python, (Nguyen et al., 2019), exhibited the best performance. It had an overall accuracy of 93% with an AUC (area under the ROC curve) of 0.95.

The model is made up of the following features (inputs) that are extracted from FullTextAnnotation and LabelAnnotation:

1. Score and Topicality for the following Labels: ‘Text’, ‘Document’, ‘Font’, ‘Paper’, ‘Paper product’, ‘Handwriting’, and ‘Writing’.

Rationale: The labels detected in the OCR output for a handwritten prescription generally contain labels such as ‘Handwriting’, ‘Writing’. Therefore, high topicality (relevance score, see Section 3.1) of such labels suggests that the prescription is handwritten.

2. Distribution of words contained in FullTextAnnotation within ten confidence score based buckets (0–0.1, 0.1–0.2, ..., 0.9–1).

Rationale: The words found in the OCR output for a handwritten prescription have, in general, lower confidence scores than those found in a printed prescription. Therefore, higher number of high-confidence words are associated with greater likelihood of a printed prescription.

3. Number of English dictionary words contained in FullTextAnnotation.

Rationale: Handwritten prescriptions generally have lesser number of words than a printed prescription since the doctor has to manually scribble while the latter is generated using a computer. Therefore, presence of a larger number of dictionary words points to the possibility of the prescription being printed.

4. Confidence scores of the languages detected by the Vision API.

Rationale: The confidence score of the detected language (English) for a handwritten prescription is generally lower than that for a printed prescription as the OCR may confuse some words as belonging to other languages. Therefore, higher confidence scores are associated with a higher likelihood of the prescription being printed.

Therefore, every prescription in the pipeline is predicted to be either printed or handwritten and based on this prediction, the medicines' names are extracted using specific procedures as depicted in [Algorithm 1](#) and [Fig. 3](#).

3.2.5.2. Brand detection model. The second model is a binary classifier that takes a word extracted from FullTextAnnotation as an input and predicts whether that word is part of a brand name of a medicine or not. We do not predict the actual brand that the word represents but only whether it is some medicine brand. This model was found necessary to process handwritten prescriptions only and its utility will be explained later, once we describe the overall algorithm.

The model was developed using the XGBoost technique, ([Chen & Guestrin, 2016](#)), and had an overall accuracy of 86.5% with an AUC (area under the ROC curve) of 0.95.

The model is made up of the following features that are extracted from FullTextAnnotation:

1. The ratio of the area of the word (found using the bounding coordinates, as described in Section 3.1) to the average area of all the words found in the prescription.

Rationale: As printed text is generally more compactly written as compared to handwritten text, this ratio is comparatively low for a printed word than a handwritten word. Since the brand detection model is applied for handwritten prescriptions only (see [Algorithm 1](#)), this feature is helpful in filtering handwritten words from the printed words. In handwritten prescriptions, the names of the medicines will be handwritten (along with some printed words from, for example, the letterhead).

2. An indicator variable which takes the value 1 if the first letter of the extracted word is in upper case and 0 otherwise.

Rationale: A medicine name in the prescription is likely to start with an upper case letter. Therefore, when this feature takes the value 1, the word is more likely to be a medicine name than when this variable takes the value 0.

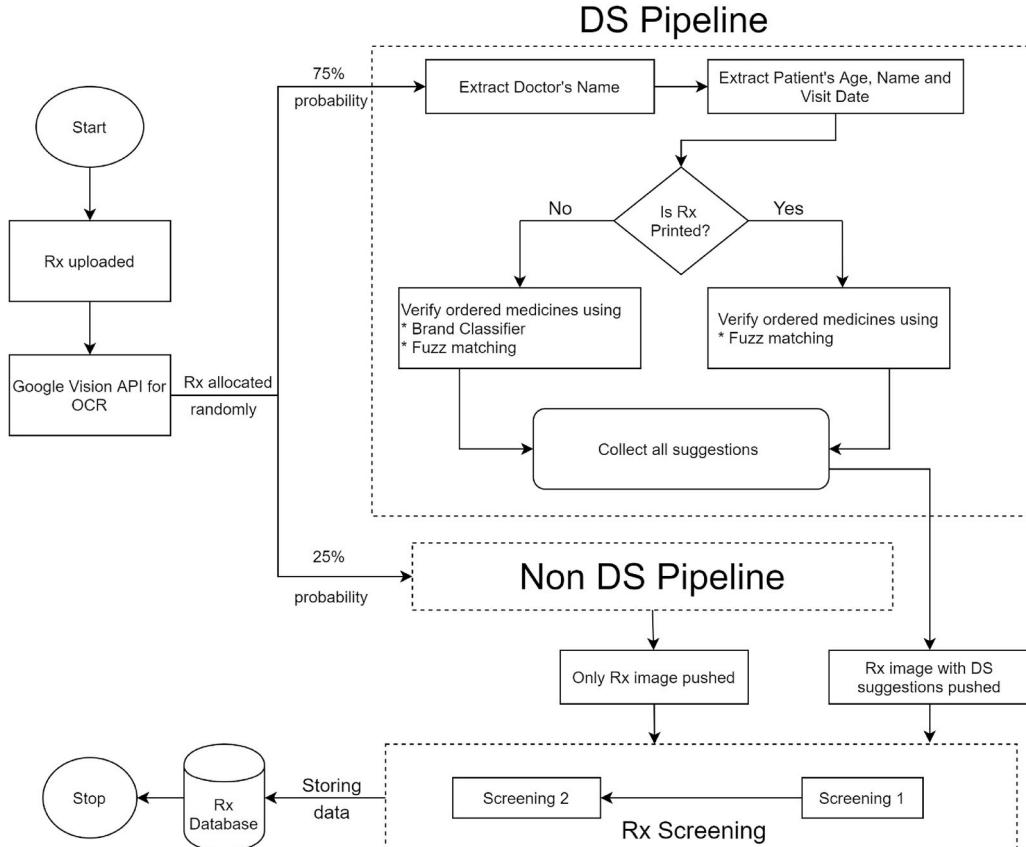


Fig. 3. The assisted AI framework for the screening process.

3. Number of letters in the word.

Rationale: It has been observed that most medicines' names are long words, generally having more than three letters. Therefore, more the number of letters in the word, more likely it is to represent a medicine name.

4. The ratio of the upper-left x-coordinate to the width of the image.

Rationale: A prescribed medicine is less likely to be present in the horizontal extremes of the prescription. This feature gives the relative position of the starting point of the word relative to the image's width. The information value obtained from the feature is that if the word is on the horizontal margins of the image, it is less likely to be a medicine name.

5. The ratio of the upper-left y-coordinate to the height of the image.

Rationale: Same reasoning as for the previous feature. A prescribed medicine is less likely to be present in the vertical extremes of the prescription image.

6. Let the *letter-density* of a word be defined as the ratio of the number of letters in a word to the area bounded by that word. Then, the last feature that was found to be significant in the model was the ratio of the letter-density of the given word to the average letter-densities of all the words found in the prescription.

Rationale: Similar to the first feature, printed words are generally more tightly written. The information value of this feature is in discerning between printed and handwritten words. Printed words will generally have above average letter-densities while the converse is true for handwritten words.

For example, if there are 200 words that are received from the OCR for a prescription, the above six features are created for each of the 200 words and the model is used to predict whether these words are a part of medicine name or not. Suppose 30 words out of the 200 words are predicted to be brand words, then only these 30 words are pushed for further downstream processing, as presented in [Algorithm 1](#).

Algorithm 1 Detection of ordered medicines in the attached prescription.

```

1: Input: List of the first names of the  $n_o$  ordered medicines,  $L_o$ 
2: Input: List of all the  $w$  words extracted from FullTextAnnotation,  $L_w$ 
3: Extract features set  $F_1$  from FullTextAnnotation
4: Extract features set  $F_2$  from LabelAnnotation
5: PrescriptionTypeClassifier( $F_1, F_2$ )
6:  $IsPrinted \leftarrow NeuralNetwork(F_1, F_2)$ 
7: return( $IsPrinted$ ) Classify the given prescription as 'Printed' or 'Handwritten'
8: end procedure
9: MedicineBrandClassifier $word$ 
10: Extract features  $F'_2 = \mathcal{F}(F_1, word)$  for the given word
11:  $IsBrand \leftarrow XGBoost(F'_2, word)$ 
12: return( $IsBrand$ ) Classify a given word as 'Medicine Brand' or 'Not a Brand'
13: end procedure
14:  $IsPrinted \leftarrow PrescriptionTypeClassifier(F_1, F_2)$ 
15: if  $IsPrinted == \text{'TRUE'}$  then
16:   for  $i = 1 \rightarrow n_o$  do
17:     for  $j = 1 \rightarrow w$  do
18:       if  $(L_o[i] \approx L_w[j]$  with fuzzy score >70%) then
19:         The ordered medicine  $L_o[i]$  has been detected in the prescription
20:       else
21:         Continue
22:       end if
23:     end for
24:   end for
25: end if
26: if  $IsPrinted == \text{'FALSE'}$  then
27:   for  $j = 1 \rightarrow w$  do
28:      $IsBrand \leftarrow MedicineBrandClassifier(L_w[j])$ 
29:     if  $IsBrand == \text{'True'}$  then
30:       for  $i = 1 \rightarrow n_o$  do
31:         if  $(L_o[i] \approx L_w[j]$  with fuzzy score >40%) then
32:           The ordered medicine  $L_o[i]$  has been detected in the prescription
33:         else
34:           Continue
35:         end if
36:       end for

```

(continued on next page)

(continued)

 Algorithm 1 Detection of ordered medicines in the attached prescription.

```

37: else IsBrand == 'False' then
38:   Continue
39: end if
40: end for
41: end if
  
```

The basic idea of the algorithm is to firstly predict whether the prescription is of type printed or handwritten. For printed prescriptions, the words extracted from FullTextAnnotation are fuzzy matched with the first names of the ordered medicines and a reasonable fuzzy score threshold of 70% helps in achieving a good balance between precision and recall.

For handwritten prescriptions, since the text is less machine readable, the words extracted from FullTextAnnotation are more deviant from the actual text than the deviation observed in the case of printed prescriptions. Therefore, direct fuzzy matching with the ordered medicines did not yield good results. To overcome this problem, we deployed two models to improve the predictive power of the system. The first model classifies every word obtained from FullTextAnnotation as either part of a medicine's brand name or a non-brand name. Then, only the words that are classified as part of medicine brand names, are fuzzy matched with the first names of the ordered medicines with a threshold of 40%. The combination of these two models was found to be much more effective than having a single model. The brand classifier model ensures that the precision of the combined system is high whereas the slightly lower threshold of 40% enables the system to have a decent recall.

3.2.6. Colored bounding boxes for DS suggestions

For every suggestion that we give from the suite of DS models for different elements of the prescription, their corresponding locations in the prescription are highlighted. As mentioned previously, Google Vision API returns four (x, y) co-ordinates for every word it finds in the document. Whenever a string in the FullTextAnnotation is determined to be an element in the prescription, these co-ordinates are returned to the front-end code of the screening tool to draw a colored box around these words in the prescription.

The suite of models described in this section is then integrated within the current process flow, as shown diagrammatically in Fig. 3.

3.2.7. Measurement of the precision of the DS suggestions

For measurement of the correctness of the suggestions, we compute two types of metrics for every prescription.

For Doctor's Name, Patient Name, Patient's Age and Visit Date we match, in a case-insensitive manner, the DS suggestion with the corresponding strings that screener enter in the screening tool. Only when there is an exact match, the DS suggestion is adjudged to be correct.

In the later section of the paper, we evaluate the impact of these DS suggestions on the order processing time for a large corpus of prescriptions, say, N .

Now, for the j^{th} prescription ($j = 1, \dots, N$), we define x_{1j} as

$$x_{1j} = \begin{cases} 1 & \text{if prediction of Doctor's Name is correct,} \\ 0 & \text{otherwise.} \end{cases}$$

Then, *percentage matched*, defined as $\bar{x}_1 = \frac{1}{N} \sum_{j=1}^N x_{1j} \times 100$ is the percentage of prescriptions in which the doctor's name was correctly predicted by the DS model.

Similarly, we define \bar{x}_i , $i = 2, 3, 4$ as the percentage matched for patient name, patient age and visit date.

For medicines, the concept is similar. As a prescription will generally contain more than one prescribed medicine, we define a different notation for the medicine related elements. Let the number of prescribed medicines in the j^{th} prescription be n_j and let $N_m = \sum_{j=1}^N n_j$ be the total number of medicines. Then,

$$x_{5j} = \begin{cases} 1 & \text{if each of the } k \text{ medicines were correctly decisioned, for all } k = 1, \dots, n_j \\ 0 & \text{otherwise.} \end{cases}$$

Then, $\bar{x}_5 = \frac{1}{N} \sum_{j=1}^N x_{5j} \times 100$ is the percentage of prescriptions in which every ordered medicine's status in the prescription was predicted correctly.

An ordered medicine is said to be decisioned correctly if the ones with the 'Verified By System' tag are not removed by the screeners and the ones which are without this tag are flagged by the screeners to be absent in the uploaded prescription.

The percentage matched metrics, \bar{x}_i , are useful in understanding the average success rate of the different methodologies in detecting the relevant elements of prescriptions. These will be discussed later in the paper.

4. Automated process: metrics framework

The above algorithms have been integrated in the current screening tool and are randomly applied to 75% of the orders flowing into the system. That is, 75% orders are assisted by the data science suggestions, whereas the remaining 25% orders are without any assistance from data science. This split enables us to measure the effectiveness of the assisted automated system against the control set.

As mentioned in the beginning of this article, the purpose of digital automation is to run high volume and repetitive tasks through software bots instead of human resources. To make a holistic assessment of the gains that accrue from an automated system (Casey, 2020), suggest the following set of metrics:

- **Productivity:** RPA should enable processes to run faster.

In our context, this means that the screening time per prescription should be lower for the assisted RPA process as compared to an unassisted process. Also, as mentioned before, the screening process is divided into S1 and S2 sections. Towards this end, we define the following test arms in the *data science (DS) pipeline*: Total time - Overall, Total time - Printed, Total time - Handwritten, S1 time - Overall, S1 time - Printed, S1 time - Handwritten, S2 time - Overall, S2 time - Printed, S2 time - Handwritten. The printed and handwritten arms, though mutually exclusive to each other, are both subsets of the parent arm, overall. Similarly, for the control group - the *non-ds pipeline* consisting of the remaining 25% of orders, the same arms are studied. Pairwise comparisons are made for these arms to measure the reduction in the processing time.

- **Consistency:** RPA should bring more consistency and predictability in the process.

To measure the change in consistency due to the assisted RPA process, we compare the standard deviations of the processing times in the DS and the non-DS pipelines.

- **Employee satisfaction:** RPA, by taking away repetitive and monotonous tasks, should result in greater employee satisfaction and experience. We assume that if the screener staff have to make fewer effort per prescription (in terms of clicks and keystroke), it would lead to higher satisfaction.

As the suggestions from the bot can be inserted by one click, employee satisfaction is expected to increase in the assisted RPA process. To measure this, we compare the number of keystrokes or clicks (defined as *total events*) required for each of the prescription elements in the assisted and unassisted process.

- **Reliability:** The automated process is expected to be more reliable with reduced downtime.

The automated process has only one external dependency: Google vision OCR. Any downtime there will directly affect the reliability of the automated process. Since the reliability of Google vision API is nearly 100%, there is no change to the degree of reliability of our process.

- **Accuracy:** RPA should enable processes to be more accurate.

Since the human agent verifies the output from the bot and signs it off, there is no machine induced inaccuracy in the process.

- **Compliance:** Automation should not diminish the compliance levels of the process.

Again, since the human agent verifies the output from the bot before signing off, the RPA assisted process is also compliant.

Therefore, the automated process described in the paper will be judged primarily on the grounds of productivity, consistency and employee satisfaction. In the next section, we present the results from our experiment which show the performance of different experiment arms in terms of productivity, employee satisfaction and consistency.

Table 1

Average order processing time (in seconds) of different arms, along with the p-values for tests of significance between different segments of the DS and non-DS pipelines.

Experiment Arm	Average Processing Time (s)		
	DS	Non-DS	p-value
Total Time - Overall	66.4	72.5	.000
Total Time - Printed	54.6	65.6	.000
Total Time - Handwritten	67.3	73.3	.000
S1 Time - Overall	32.8	34.4	.000
S1 Time - Printed	30.4	36.1	.000
S1 Time - Handwritten	39.1	41.2	.000
S2 Time - Overall	28.2	31.8	.000
S2 Time - Printed	24.2	29.5	.000
S2 Time - Handwritten	28.2	32.1	.000

5. Results

The dataset that we use for analysis in this paper consist of 115,727 prescription orders which are about a month's orders. Of these orders, 81.6% have one prescription attached, 15.6% have two prescriptions and 2.8% orders have more than two prescriptions attached.

Of all the prescriptions attached with these orders, 81.8% are of the type handwritten and rest 18.2% are printed.

As mentioned before, about 75% (86,641) of the orders are serviced through the DS pipeline and the rest 25% (29,086) orders are serviced through the non-DS pipeline.

5.1. Evidence of increased productivity

As mentioned in the previous section, we compare the average processing time between the DS and the non-DS pipelines. This comparison is studied for the overall process as well as S1 and S2 sections of the screening process. Within these arms, we further compare the set of printed and handwritten prescriptions.

To test for significance of the differences between the average processing times between the DS and non-DS pipelines, we apply Welch's test, a two-tailed *t*-test to test the significance of difference of two independent means, (Welch, 1947). Table 1 presents the average processing time in the nine experimental arms and the associated p-values.

It can be seen from the table that all arms within the DS pipeline have lower average processing times as compared to the same arms in the non-DS pipelines. Furthermore, the differences in the averages were found to be statistically significant. The assisted RPA process run, on an average 8.4% faster than the unassisted process. A printed prescription, on average, gets processed 16.8% faster in the DS pipeline as compared to the non-DS pipeline. On the other hand, a handwritten prescription gets processed 8.2% faster in the DS pipeline. As mentioned before, since printed prescriptions are only about 18.2% of the total prescriptions, the bigger gains in efficiency from the printed prescriptions get scaled down in the overall pipeline.

To give a greater insight into the distributions of the process times, Fig. 4 presents the percentile distributions of the order processing times for four different categories of prescriptions in the DS and non-DS pipelines.

It can be seen from the figures that the separation between the DS and non-DS pipeline is higher in the case of printed prescriptions.

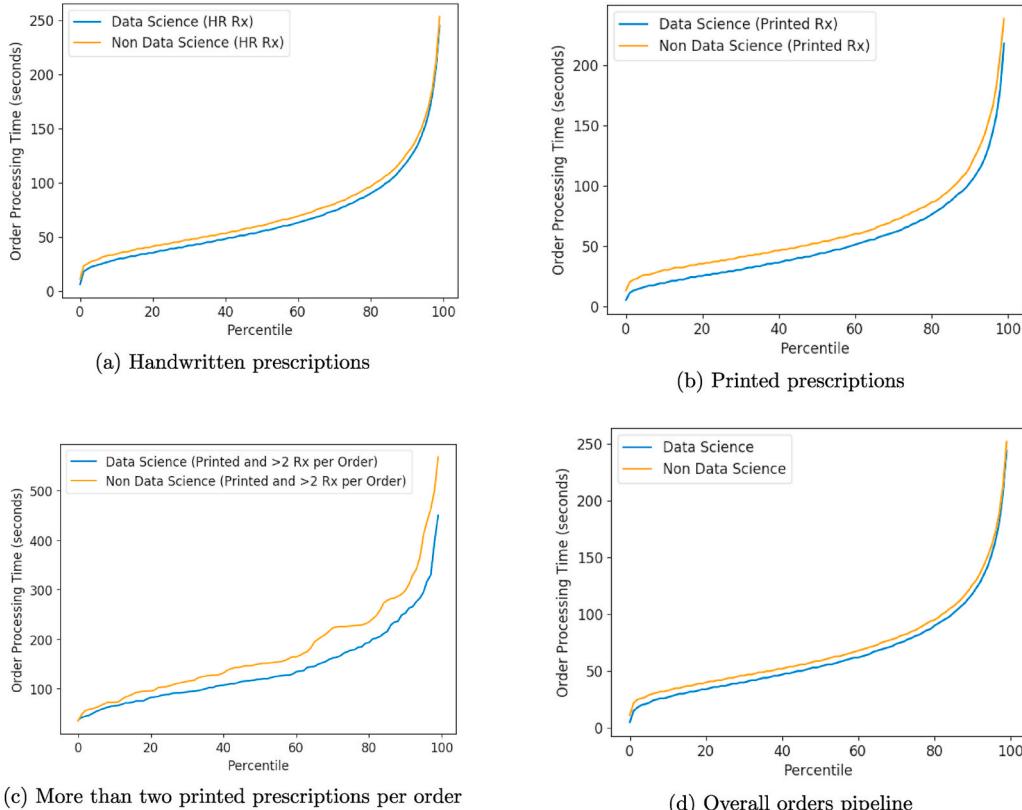


Fig. 4. Percentile distributions of the order processing times in the DS and non-DS pipelines for four categories of orders: orders containing handwritten prescriptions, containing printed prescriptions, containing more than two printed prescriptions, and overall orders pipeline. The x-axes represent the percentile number and the y-axes represent the corresponding value of the percentile.

As mentioned previously, about 2.8% of orders have more than two prescriptions with them. For such orders, when the attached prescriptions are all printed, the differences between the processing times in DS and non-DS pipelines are much more pronounced.

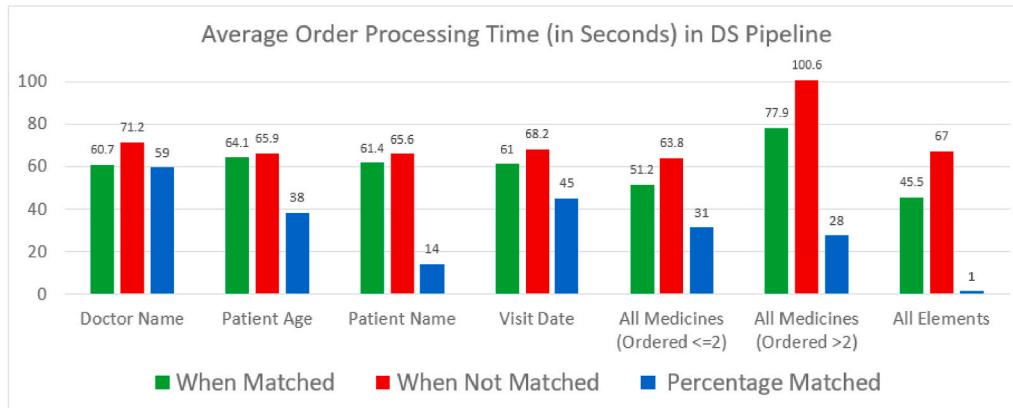
Next, we see the attribution of the savings in the processing times to the correct suggestions by the DS algorithms defined in Section 3.2. The reduction in the processing times in DS pipeline are account of correct suggestions that get selected in the screening tool. Therefore, greater the number of correct suggestions by the DS algorithm, the lower the processing time of that prescription is expected to be. To investigate this further, we present, in Fig. 5, the average processing times in the DS pipeline when different elements of the prescription are correctly matched with the suggestions from our models.

It can be seen from Fig. 5a that in the overall pipeline, the doctor's name is correctly predicted in 59% of the prescriptions. When the doctor's name is correctly matched, the average processing time of an order is 60.7 s versus 71.2 s when the suggestion is incorrect (irrespective of match status of other elements in the prescription), that is, a saving of about 15% in the average order processing time. Similarly, savings in average processing times and the percentage matches can be observed for patient age, patient name and visit date.

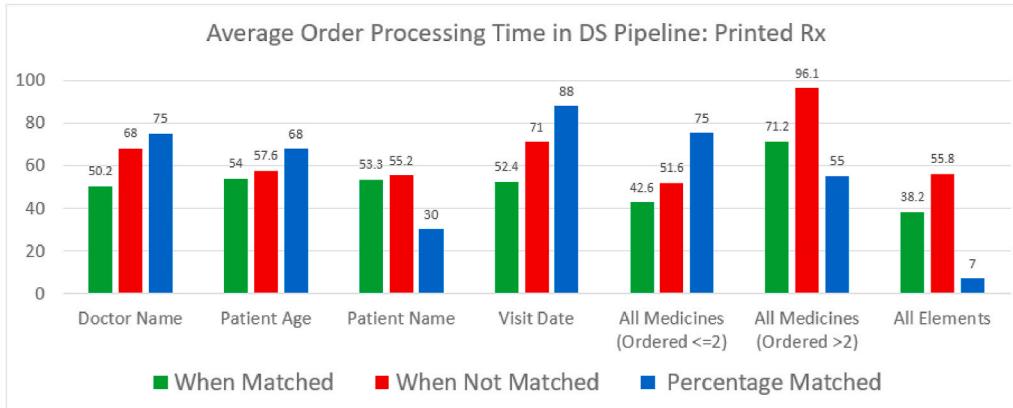
Since the number of ordered medicines per prescription is variable, we present the metrics for medicines in two separate categories. The first category is for orders where the number of ordered medicines is up to two - such orders are about 73% of the total orders. The second category, which consists of about 23% of the total orders, have at least three medicines ordered.

It can be seen that when all the ordered medicines are correctly predicted by the DS algorithms, the average processing time reduces by more than 20%. Furthermore, in about 1% of orders in the DS pipeline, our models are able to predict every element of the prescription correctly - for such orders, reduction in the average processing time because of the data science assistance is in excess of 32%.

In Fig. 5b, we present the same data, but only for printed prescriptions in the DS pipeline. The match percentage of the elements is significantly higher in case of printed prescriptions, which is expected, as the printed text's machine readability is expected to be higher than that of the handwritten text. Doctor's name, which is correctly predicted for 59% of the prescriptions overall, is matched



(a) Overall DS pipeline



(b) Printed prescriptions in the DS pipeline

Fig. 5. Average processing times in the DS pipeline when different elements of the prescription are correctly matched with the suggestions from the data science models described in Section 3.2. The medicines metrics are presented in two separate categories - in the first, the number of ordered medicines are is two and the second category is for orders where the number of ordered medicines exceeds two.

correctly in 75% of printed prescriptions. Furthermore, the reduction in the average processing time because of a correct match of doctor's name, increases from 15% to more than 26%. Similarly, for other elements of a prescription, the match percentages nearly double for printed prescriptions.

Furthermore, in about 7% of the printed prescriptions, all elements are correctly predicted by the data science models. One of the end-objectives of our automated process is to achieve 100% automation in the case of printed prescriptions. We discuss this briefly at a later place in the paper. We analyze the consistency or the variability in the processing times in both pipelines later in Section 5.3.

Thus, we establish in this section that the assisted RPA process, on the strength of predictive models, runs significantly faster as compared to the unassisted process. In the next section, we show that the assisted automation also leads to reduction in the effort that the screener staff have to invest per prescription in terms of the number of keystrokes and mouse-clicks.

5.2. Evidence of enhanced employee satisfaction

As mentioned in Section 2, the screening staff have to locate and identify the relevant elements in the prescription and enter them in the S1 and S2 sections of the screening tool, as shown in Fig. 1. To facilitate the recording of these data in our databases via the screening tool, a screener adjusts the image position through clicking and zooms in or out. Furthermore, once the relevant element of a prescription has been identified, the data point has to be entered in the screening tool. Therefore, if a screener event is defined as an instance of a screener using her computer's mouse or keyboard within the screening tool. The following types of screener events can be defined in our application:

- Mouse-clicks events (for image manoeuvring and to initiate keyboard input): On prescription image for moving it, zoom in button, zoom out button, doctor's name text box, patient's name text box, patient's age text box, and visit date box.
- Keystroke events (for entering data in the tool): doctor's name, patient's name, patient's age, and visit date.

For medicines, there are no events required to be recorded as there is no data which the screener has to enter in the tool. The prescription only has to be vetted for the presence of the medicines contained in the ordered list.

As mentioned in Section 1, one of the expectations from RPA is that it leads to greater employee engagement, by taking over monotonous and repetitive tasks. In our context, the data science enabled highlighted boxes is expected to reduce the manoeuvring (such as click on the image and zoom ins and outs, as described above) a screener has to do within the prescription to locate the relevant fields, and the clickable predictive suggestions from the models are expected to reduce the keystrokes in these text boxes. We henceforth refer to the mouse-clicks and keystrokes by a screener on a prescription as *total screener events*. Therefore, by measuring the difference in the average total screener events in DS and non-DS pipeline, the contribution of assisted RPA in reducing manual effort, and thereby increasing employee satisfaction, can be ascertained.

Towards this end, we record the number of total screener events for every prescription in both DS and non-DS pipelines. In Table 2, we present the average number of total screener events for different segments in the two pipelines. Statistical significance of the difference in means is tested using Welch's test.

From the table, it can be seen that the average number of total screener events is lower in the DS pipeline as compared to the non-DS pipeline and this difference is statistically significant.

To understand the inter-relationships between the prediction success of various elements in a prescription, the processing time and the number of total screener events, we present the correlation matrix for these variables in Fig. 6. As mentioned before, the variables related to doctor's name, patient's age, patient's name and visit date each take value 1 if the corresponding suggestions given by DS algorithms are accepted by the screener staff and 0 otherwise. Total time denotes the processing time of a prescription and total events denotes the number of total screener events.

There are a few interesting observations that can be drawn from the correlation matrix. Firstly, it can be seen that, in general, whenever elements of a prescription are correctly matched, it leads to a reduction in the processing time and the total number of screener events, as seen by the negative correlations (except for weak positive correlation in the case of patient age and patient name). Furthermore, when all the ordered medicines are matched correctly, it has the highest impact on reduction in processing time, as indicated by a correlation coefficient of -22%. On the other hand, since there are no screener events associated with medicines' screening, the variable measuring whether all medicines were matched has the lowest correlation coefficient.

As expected, there is a positive correlation (16%) between the number of total screener events and the prescription processing time.

Matching of patient name, patient age and the visit date have strong positive pair-wise correlations of greater than 20%. This means that these three elements of the prescription are more likely to be predicted correctly together. However, prediction of doctor name,

Table 2

Average number of screener events of different arms, along with the p-value of test of significance between the data science and non-data science processing time.

Average Number of Total Screener Events	DS	Non-DS	p-value
Experiment Arm			
Overall	27.9	29.3	.000
Printed	23.9	25.7	.000
Handwritten	28.8	30.2	.000



Fig. 6. Correlation matrix for prediction successes of various elements in a prescription, the order processing time and the number of total screener events.

while having a strong negative correlation with the prescription processing time and total screener events, has a weaker positive correlation with the predictabilities of the above three elements. A plausible reason for this is that the doctor's name is always printed, even in handwritten prescriptions, and therefore, its match rate in handwritten prescriptions is considerably higher than other elements (refer to Fig. 5a). Because of this dissonance, the correlation of prediction of doctor's name is weaker with other elements.

Thus, we establish in this section that the assisted RPA process, on the strength of predictive models, requires fewer effort by the screener staff. In the next section, we compare the consistency or the variability in the assisted RPA process and the unassisted process.

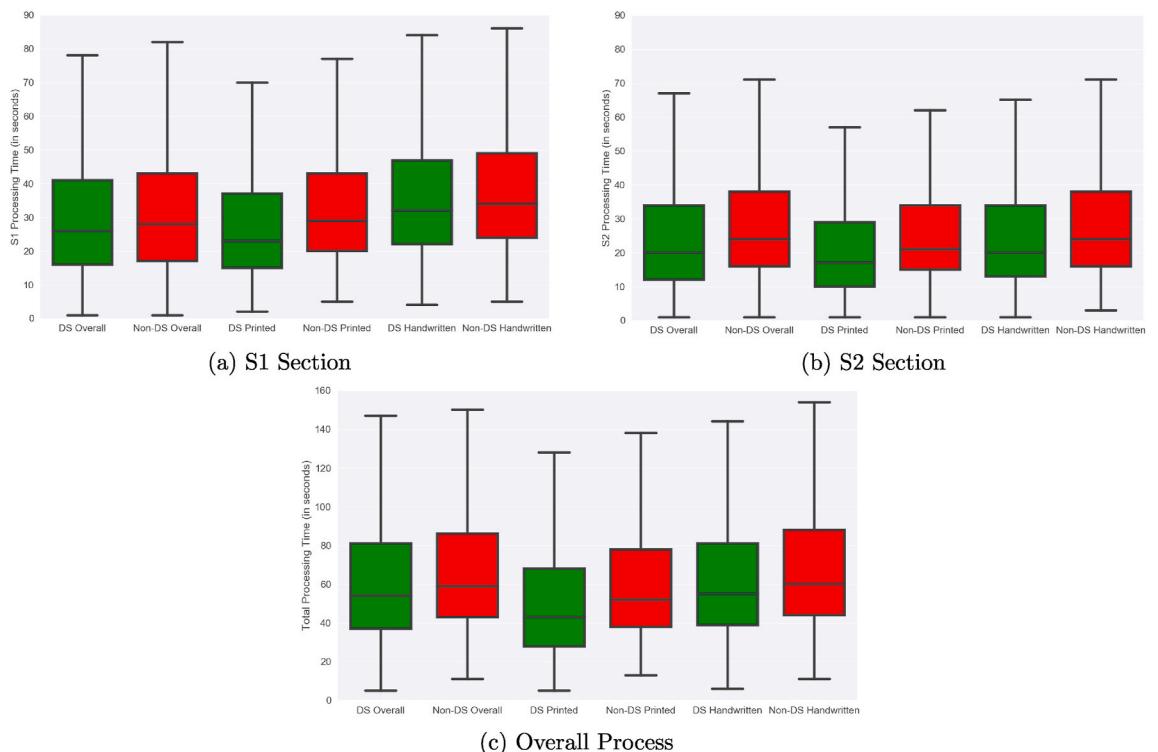


Fig. 7. Boxplot distributions of the order processing times for different segments in the DS and non-DS pipelines.

5.3. Evidence of enhanced consistency

Measurement and control of process variation is central to maintenance of quality in a process. A consistent process behaves in a predictable way and is, therefore, reliable and cost-effective. Process variation is of two kinds: natural or uncontrollable variation and special-cause or controllable variation, (Bowen & Neuhauser, 2013). While the former is intrinsic to the process and, generally, cannot be controlled, the latter is because of an external cause that can usually be determined and mitigated.

RPA is expected to reduce overall variability in the process or, at least, make the process run faster without increasing the existing process variation. In the previous two sections, we showed that the assisted RPA runs faster and requires fewer effort from the screening staff. We now investigate if the assisted process also has lower variability in the processing times and total screener events.

In Fig. 7, we present the boxplot distributions of the processing times for different segments in the DS and non-DS pipelines. The boxplots that we present in this paper have the box made up of the 3rd and 1st quartile, i.e., Q3 and Q1 with the median as the line inside the box. The lower whisker is at the lowest data point above $Q1 - 1.5(Q3 - Q1)$ and the upper whisker at the highest data point below $Q3 + 1.5(Q3 - Q1)$.

It can be seen from the figures that, in general, the DS pipeline has lower spread in all its segments, and in particular, in the S2 section of the process and for printed prescriptions.

To formally test the significance of differences between the standard deviations of the processing times in different segments of the DS and non-DS pipelines, we apply the Bartlett's test for homogeneity of variances, (Bartlett & Fowler, 1937). The segments' standard deviations and the p-values are presented in Table 3.

It can be seen from the table that except for the handwritten prescriptions in the S1 section, the standard deviations of the segments in the DS pipeline are lower than those in the non-DS pipeline and these differences are statistically significant. Furthermore, it appears that the assisted RPA process for the printed prescriptions is especially more consistent than the printed prescriptions in the non-DS pipeline.

Next, we investigate the effect of assisted RPA on the variability in the screeners' effort. Fig. 8 presents boxplot distributions of total screener events for different segments in the DS and non-DS pipelines.

The spread for the segments DS pipeline segments looks slightly lower as compared to non-DS pipeline's segments.

Again, to formally test the significance of differences between the standard deviations of the different segments of the DS and non-DS pipelines, we apply the Bartlett's test. The segments' standard deviations and the associated p-values are presented in Table 4.

It can be seen from the table that the assisted RPA process does lead to a reduction in the variability in the efforts that the staff make in processing an order. As before, printed prescriptions have lower variability in the screener effort as compared to handwritten prescriptions.

Thus, we establish that the DS pipeline process prescriptions faster, with fewer effort and with lower variability and hence demonstrate the value of the assisted RPA process.

6. Applications of digitized prescriptions' data

In this section, we discuss some of the benefits and applications of the data garnered from digitization of prescriptions in terms of their utility to not only the business but also to public healthcare and medical research community. These applications underpin the importance of the assisted RPA process in making our digitization process more efficient and scalable.

6.1. Healthcare trends and business analytics

While standard business intelligence systems maintain sales data in most industries, a prescription provides a unique opportunity to a healthcare services provider for understanding their clients' needs and past history. A prescription is a snapshot in time of a customer's health - it gives deep insights into the current health status of an individual. While the medicines that are written on the prescription could also be fetched from the sales databases, it is the accompanying information on the prescription such as age, gender,

Table 3

Standard deviations of processing times of different segments, along with the associated p-values of tests of significance for the DS and non-DS pipelines.

Variability in Processing Times: Standard Deviation			
Experiment Arm	DS	Non-DS	p-value
Total Time - Overall	46.0	46.9	.000
Total Time - Printed	41.6	45.7	.000
Total Time - Handwritten	45.5	46.7	.000
S1 - Overall	28.2	28.6	.016
S1 - Printed	25.3	26.8	.000
S1 - Handwritten	28.0	28.4	.088
S2 - Overall	27.9	28.9	.000
S2 - Printed	25.0	29.1	.000
S2 - Handwritten	27.9	29.0	.000

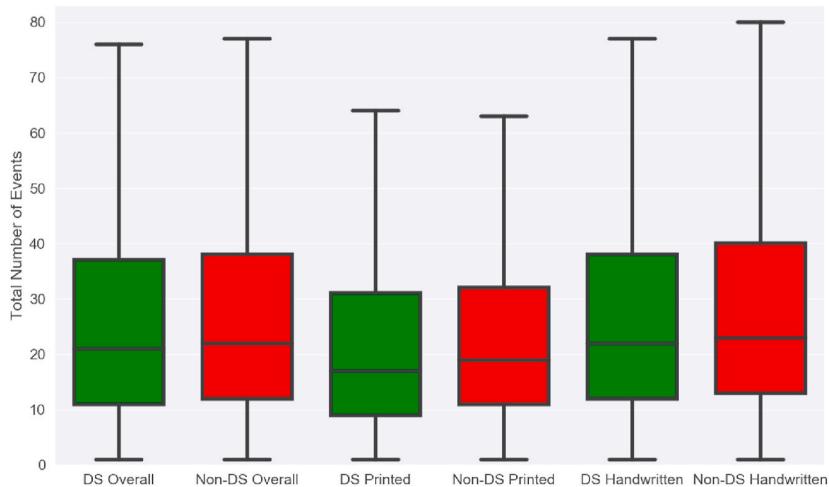


Fig. 8. Boxplot distributions of total screener events for different segments in the DS and non-DS pipelines.

Table 4

Standard deviations of total screener events of different segments, along with the associated p-values of tests of significance for the DS and non-DS pipelines.

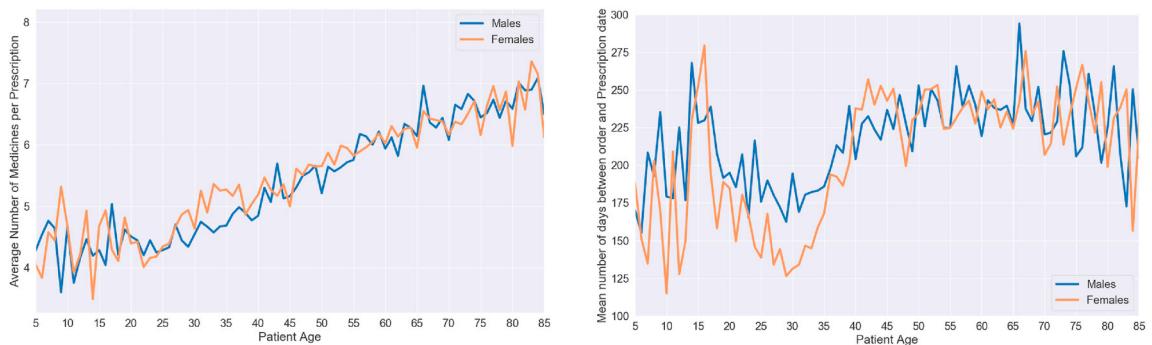
Total Screener Events Variability: Standard Deviation			
Experiment Arm	DS	Non-DS	p-value
Overall	25.3	26.3	.000
Printed	23.4	24.5	.000
Handwritten	25.6	26.6	.000

dosage, dosing frequency, appointment date and symptoms which paint a wholesome picture of a patient's health and help put the medicines' information into perspective.

For example, Fig. 9 presents two age-wise plots of the data from our prescription data.

Fig. 9a presents the average number of medicines per prescription with patient age for both genders. After the age of 25 years, the average steadily increases with age for both genders and the movement in this metric is similar for both genders. However, between the ages 27 and 37, female patients exhibit a markedly higher average of medicines per prescription. A plausible reason for this phenomenon could be that many of the female patients in this age-group experience pregnancies due to which they require more medicines, on average, than the males in the same age group.

Fig. 9b presents the distribution of the mean number of days between the order placement date and the doctor visit date with the age of the patient. That is, the lag between the date the prescription is written and the date when an order is placed based on that prescription. In general, the lag is smaller for young adults and increases up to age 45 years, possibly because of the gradual onset of chronic diseases. It stabilizes between ages 45 and 75 years, and starts declining after this age, possibly because of the emergence of



(a) Average number of medicines per prescription.

(b) Mean lag between dates of order and visit.

Fig. 9. Insights from the prescriptions' data: (a) Distribution of the average number of medicines per prescription with patient age. (b) Distribution of the mean number of days between the order date and the doctor visit date with patient age.

acute disease in older age.

Similarly, the mean age for onset of different diseases can be estimated through the prescriptions' data, which is representative, to a large extent, of the general population. For example, Fig. 10 presents the mean age of patients suffering from different forms of cancer, as deduced by the medicines prescribed to them.

Aggregation of individual patient's information can give interesting insights and help businesses in pre-empting the needs of customers. These insights can be used to form cohorts or segmentations of the customer base which can then be targeted intelligently by customized recommendations and campaigns.

6.2. Disease models and post-marketing surveillance

The medicines in a prescription, their recommended duration of consumption, frequency of ingestion and dose sizes, along with the patient information such as age and gender are not just a reflection of a patient's current health status but also provide valuable signals of the trajectory of current and future diseases. In fact, models provided by, e.g. (Dehkordi & Sajedi, 2019), enable determination of the diseases on the basis of the information contained in the prescription.

In our context, the large corpus of digitized prescriptions can serve two purposes. Firstly, the inferred diseases and the prescribed medicines can be tracked to pick signals of emerging patterns before they become strong trends and can pose community health risk, (Agrebi & Larbi, 2020; Davenport & Kalakota, 2019). Such signals can be used as inputs in larger epidemiology intelligence systems to forewarn about emergence of novel health risks, so that pre-emptive actions can be undertaken, (Wilburn et al., 2019).

Secondly, given that we have a large number of repeat users on the platform, longitudinal studies can be implemented on patients having a characteristic of interest to understand disease trajectories and develop disease models, (Caruana et al., 2015; Wang et al., 2014). For example, an increase (decrease) in the dosage or dosing frequency with time for a patient signifies worsening (improvement) of the disease. Such patients can be used to form cohorts which can then be analysed to identify disease specific risks such as age, co-morbidities, and co-prescribed medicines to develop risk models for disease prediction.

While information garnered from the drugs a patient is prescribed over time can help in understanding the progression of a disease and general health status, for newly launched drugs, the converse is also true. That is, the progressive effect of consuming a particular drug on the quality of life of the patients can also be studied over a period of time.

The drug developmental process is time consuming and very expensive, with an estimated average cost of development of a new prescription drug to be up to \$2.8 billion, (DiMasi et al., 2016). There are typically four phases in a clinical trial, with every phase involving testing of the drug on a larger cohort of subjects than the previous phase.

Although the trial process is designed to be robust and is supported by scientifically sound decision criteria, it involves subjects in a trial setting only and may not necessarily reflect the behaviour of the drug in real world scenario and for a large number of patients. Towards this end, post-marketing surveillance (PMS) - the practice of monitoring the safety of a drug after it has been released on the market - ensures that the drug continues to be safe in the general population. There are multiple ways of carrying out PMS - spontaneous reporting databases, drug registries and patient health records, (McNeil et al., 2010). In our context, studies can be undertaken, in collaboration with pharmaceutical companies, which track patients consuming the specific drug under consideration and compare their prescriptions' data with patients having similar risk factors but not consuming this drug, to draw out the safety profile of this newly marketed drug.

7. Conclusions and further work

In this paper, we have demonstrated that the assisted RPA enables our prescription validation and digitization process to run faster and with more consistency. The reduction in the processing time helps us to save more about 2100 man-hours every year which is a decent cost saving. Furthermore, by reducing the effort of the screening staff, we are able to enhance employee experience as well. These efficiencies add not just business value, but as discussed previously, catalyse the efforts of medical research.

Our end-goal is to develop a 100% autonomous system which processes a prescription with no human intervention. However, because of compliance reasons, such a system would still require some human overseeing to ensure data quality and system integrity. Towards this end, the system would need to be designed on the lines of a financial fraud detection system, wherein the predictive system flags suspicious transactions and imposes a higher level of verification and authentication checks on them, (John et al., 2016). In our context, when it is predicted that the data science models may not have predicted a prescription element correctly (by a separate model which considers features such as prescription type, OCR confidence scores and previous training data), it is pushed for a screener to verify. Such an arrangement will significantly reduce the time taken per prescription further.

A pre-requisite to achievement of the above goal is greater accuracies (percentage matched) of our data science models suite. We intend to pursue two directions for this improvement.

The first direction is to leverage the co-existence tendencies of various medicines. Association rules mining, which consists of analysing data for patterns, or co-occurrence and identification of frequent if-then associations, called association rules. Often, medicine pairs tend to occur together, even when they might be belonging to different classes. For example (Sutherland et al., 2015), found that the beta-blocker metoprolol and anti-platelet agent clopidogrel occurred 3 times more than expected if they were used independently which is consistent with their use for treating cardiovascular disease. Such association rules can be very useful in improving the recall of the medicine detection algorithm. That is, if one medicine in the ordered list of medicines is found in the prescription, and there is another ordered medicine which is known to co-occur with the found medicine by an association rule, it is highly probable that the second medicine would also be present in the prescription. Similarly, the regular expression or named entity

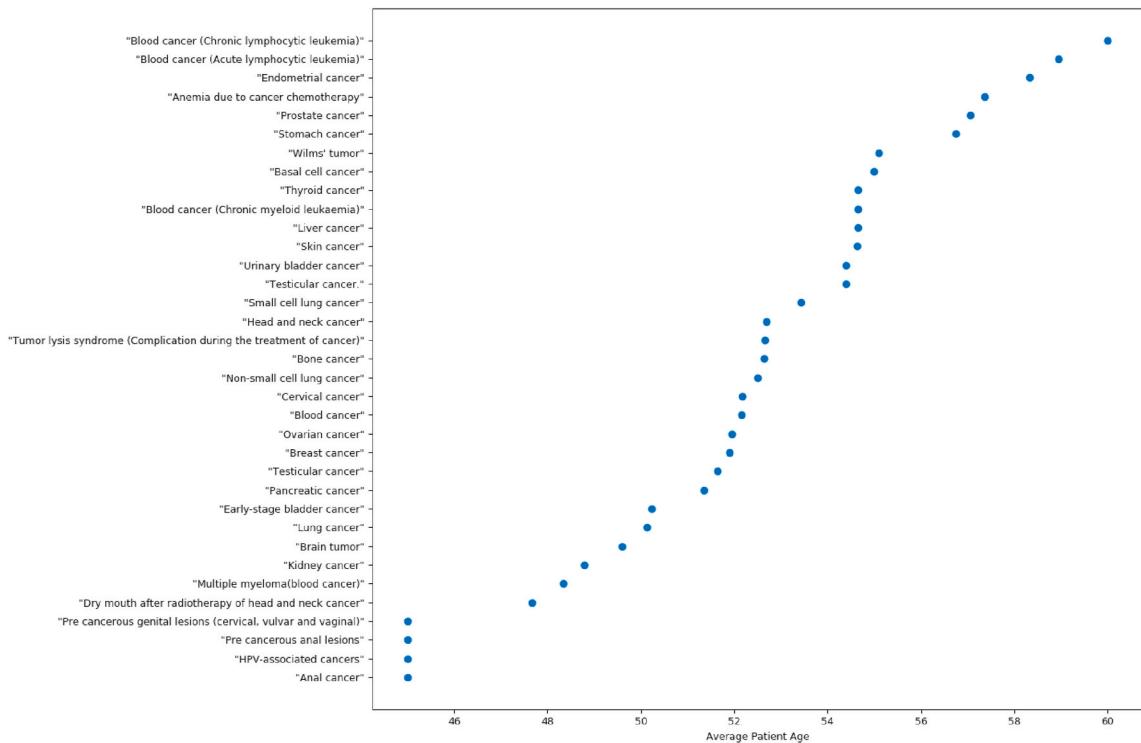


Fig. 10. Average ages of patients suffering from different cancers or suffering from side-effects of anti-cancer treatments.

recognition methods discussed in Section 3.2 can be used to detect the specialization of the doctor that had written that prescription. Similar to association rules, the detected specialization can be used to re-prioritise the search space for the medicines.

The second direction is pre-processing of images before sending them to the Google Vision API. Currently, the files uploaded by the customers are sent to the Vision API without any image pre-processing from our side. The files supplied by the customers have a large variance in the image quality. As shown by Hosseini et al. (2017), the Vision API is not immune to image noise and even minor deteriorations in the image quality can significantly reduce the accuracy of the API. Therefore, pre-processing actions such as cropping, gray-scaling and blur reduction could possibly enable better quality output from the Vision API, thereby improving the accuracy of our model suite and further reduction in the processing times. We intend to work in these two directions in the future.

Credit author statement

Kabir Soeny: Conceptualization, Formal analysis, Writing – original draft, Visualization, Project administration. Gaurav Pandey: Methodology, Software, Data curation, Validation. Utkarsh Gupta: Methodology, Software, Data curation. Adarsh Trivedi: Methodology, Software. Mehul Gupta: Methodology, Software, Writing – review & editing. Gaurav Agarwal: Resources, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors are grateful to the anonymous reviewers whose suggestions have led to an improved manuscript.

References

- 1mg Technologies, 2020 1mg Technologies. (2020). *1mg - online medical store and healthcare platform*. www.1mg.com.
- Agrebi, S., & Larbi, A. (2020). Use of artificial intelligence in infectious diseases. *Artificial Intelligence in Precision Health*, 415–438.
- Anywhere, A. (2020). What's the difference between attended and unattended rpa?. <https://www.automationanywhere.com/rpa/attended-vs-unattended-rpa>.
- Aronson, J. (2009). Medication errors: What they are, how they happen, and how to avoid them. *QJM: International Journal of Medicine*, 102(8), 513–521.
- Asquith, A., & Horsman, G. (2019). Let the robots do it! - taking a look at robotic process automation and its potential application in digital forensics. *Forensic Science International: Reports*, 1, 100007.

- Bartlett, M. S., & Fowler, R. H. (1937). Properties of sufficiency and statistical tests. In *Proceedings of the royal society of London. Series A - mathematical and physical sciences* (Vol. 160, pp. 268–282), 901.
- Bowen, M. E., & Neuhauser, D. (2013). Understanding and managing variation: Three different perspectives. *Implementation Science*, 8(1), S1.
- Caban, J. J., Rosebrock, A., & Yoo, T. S. (2012). Automatic identification of prescription drugs using shape distribution models. In *2012 19th IEEE international conference on image processing* (pp. 1005–1008).
- Caruana, E. J., Roman, M., Hernández-Sánchez, J., & Solli, P. (2015). Longitudinal studies. *Journal of Thoracic Disease*, 7(11), E537–E540.
- Casey, K. (2020). *Robotic process automation (rpa) metrics: How to measure success*. <https://enterprisersproject.com/article/2019/6/rpa-robotic-process-automation-metrics-how-measure-success>.
- Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 785–794). New York, NY, USA: Association for Computing Machinery.
- Davenport, T., & Kalakota, R. (2019). The potential for artificial intelligence in healthcare. *Future Healthcare Journal*, 6(2), 94–98.
- Dehkordi, S. K., & Sajedi, H. (2019). Prediction of disease based on prescription using data mining methods. *Health Technology*, 9(1), 37–44.
- DiMasi, J. A., Grabowski, H. G., & Hansen, R. W. (2016). Innovation in the pharmaceutical industry: New estimates of r&d costs. *Journal of Health Economics*, 47, 20–33.
- Fajardo, L. J., Sorillo, N. J., Garlit, J., Tomines, C. D., Abisado, M. B., Imperial, J. M. R., Rodriguez, R. L., & Fabito, B. S. (2019). Doctor's cursive handwriting recognition system using deep learning. In *2019 IEEE 11th international conference on humanoid, nanotechnology, information technology, communication and control, environment, and management (HNICEM)* (pp. 1–6).
- Google. (2020). *Vision AI*. <https://cloud.google.com/vision>.
- Hosseini, H., Xiao, B., & Poovendran, R. (2017). *Google's cloud vision api is not robust to noise*.
- John, S., Anele, C., Kennedy, O., Olajide, F., & Kennedy, C. (2016). Realtime fraud detection in the banking sector using data mining techniques/algorithm. In *2016 international conference on computational science and computational intelligence (CSCI)* (pp. 1186–1191). IEEE Computer Society.
- Madakam, S., Holmukhe, R. M., & Jaiswal, D. K. (2019). The future digital work force: Robotic process automation (RPA). *JISTEM - Journal of Information Systems and Technology Management*, 16.
- McNeil, J. J., Piccenna, L., Ronaldson, K., & Ioannides-Demos, L. L. (2010). The value of patient-centred registries in phase iv drug surveillance. *Pharmaceutical Medicine*, 24(5), 281–288.
- Neunerdt, M., Trevisan, B., Reyer, M., & Mathar, R. (2013). Part-of-speech tagging for social media texts. In I. Gurevych, C. Biemann, & T. Zesch (Eds.), *Language processing and knowledge in the web* (pp. 139–150). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., López García, Á., Heredia, I., Malík, P., & Hluchý, L. (2019). Machine learning and deep learning frameworks and libraries for large-scale data mining: A survey. *Artificial Intelligence Review*, 52(1), 77–124.
- Porterfield, A., Engelbert, K., & Coustasse, A. (2014). Electronic prescribing: Improving the efficiency and accuracy of prescribing in the ambulatory care setting. *Perspectives in health information management*, 11, 1g.
- Shepard, D. H. (1951). *David shepard invents the first ocr system "gismo"*. <https://www.historyofinformation.com/detail.php?entryid=885>.
- Sutherland, J. J., Daly, T. M., Liu, X., Goldstein, K., Johnston, J. A., & Ryan, T. P. (2015). Co-prescription trends in a large cohort of subjects predict substantial drug-drug interactions. *PLoS One*, 10(3), 1–19.
- Thompson, K. (1968). Programming techniques: Regular expression search algorithm. *Communications of the ACM*, 11(6), 419–422.
- Ukkonen, E. (1985). Algorithms for approximate string matching. *Information and Control*, 64, 100–118.
- Van Der Aalst, W. M. P., Bichler, M., & Heinzl, A. (2018). Robotic process automation. *Business & Information Systems Engineering*, 60(4), 269–272.
- Wang, X., Sontag, D., & Wang, F. (2014). Unsupervised learning of disease progression models. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 85–94). Association for Computing Machinery.
- Welch, B. L. (1947). The generalization of Student's problem when several different population Variances are involved. *Biometrika*, 34, 28–35.
- Wilburn, J., O'Connor, C., Walsh, A. L., & Morgan, D. (2019). Identifying potential emerging threats through epidemic intelligence activities - looking for the needle in the haystack? *International Journal of Infectious Diseases*, 89, 146–153.
- Yadav, V., & Bethard, S. (2018). A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th international conference on computational linguistics* (pp. 2145–2158). Santa Fe, New Mexico, USA: Association for Computational Linguistics.