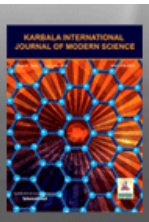

Karbala International Journal of Modern Science

Volume 9 | Issue 2 Article 11

Analyzing COVID-19 Vaccine Adverse Reactions Using Machine Learning Techniques

Mohammed Basil Albayati
Department of Computer Science, College of Computer Sciences and Mathematics, Tikrit University, Tikrit, Iraq, moh.b.m@tu.edu.iq

Ahmad Mousa Altamimi
Department of Software Engineering, Princess Sumaya University for Technology, Amman, Jordan

Recommended Citation

Albayati, Mohammed Basil and Altamimi, Ahmad Mousa (2023) "Analyzing COVID-19 Vaccine Adverse Reactions Using Machine Learning Techniques," *Karbala International Journal of Modern Science*: Vol. 9 : Iss. 2 , Article 11.
Available at: https://doi.org/10.33640/2405-609X.3271

Analyzing COVID-19 Vaccine Adverse Reactions Using Machine Learning Techniques

Abstract

COVID-19 vaccination helps protect people from getting the virus. Some people show up normal signs from the vaccine, which indicates that their body is building protection. However, adverse effects on people could cause long-term health problems. Severe allergic reactions, Myocarditis, and Pericarditis appeared in the vaccinated people that have been reported to the (FDA/CDC) Vaccine Adverse Event Reporting System (VAERS). In fact, other possible effects are still being studied in clinical trials. In the present work, the adverse reactions caused by Covid-19 vaccines of Pfizer/BioNTech, Moderna, and JJ Johnson & Johnson manufacturers are studied. Specifically, the supervised machine learning approach is utilized to discriminate body reactions against the vaccine and provide a decision-making model for the vaccine recipients. The model study and analyze the recipients' reactions whether they showed mild, moderate, or severe acute syndromes to reduce the fatality rates. To validate our model, a dataset of more than 52k records with 18 informative attributes provided by VAERS has been utilized, and three supervised learning algorithms have been implemented in Python which are Decision Tree, Support Vector Machine, and Naïve Bayes to conduct two experiments. A simple splitting percentage method was performed in the first one, while a k-Folds Cross-validation technique was used in the second experiment with k=5. The model showed a promising result with stable performance in both experiments, the Decision Tree outperformed other algorithms with a predictive rate of 0.91999 in the first experiment, and 0.91369 in the second one.

Keywords

Covid-19, Vaccine, Adverse Reactions, Machine Learning (ML), Pfizer/BioNTech, Moderna, Johnson & Johnson

RESEARCH PAPER

Analyzing COVID-19 Vaccine Adverse Reactions Using Machine Learning Techniques

Mohammed B. Albayati [a,*], Ahmad M. Altamimi [b]

[a] Department of Computer Science, College of Computer Sciences and Mathematics, Tikrit University, Tikrit, Iraq
[b] Department of Software Engineering, Princess Sumaya University for Technology, Amman, Jordan

Abstract

COVID-19 vaccination helps protect people from getting the virus. Some people show up normal signs from the vaccine, which indicates that their body is building protection. However, adverse effects on people could cause long-term health problems. Severe allergic reactions, Myocarditis, and Pericarditis appeared in the vaccinated people that have been reported to the (FDA/CDC) Vaccine Adverse Event Reporting System (VAERS). In fact, other possible effects are still being studied in clinical trials. In the present work, the adverse reactions caused by Covid-19 vaccines of Pfizer/BioNTech, Moderna, and JJ Johnson & Johnson manufacturers are studied. Specifically, the supervised machine learning approach is utilized to discriminate body reactions against the vaccine and provide a decision-making model for the vaccine recipients. The model study and analyze the recipients' reactions whether they showed mild, moderate, or severe acute syndromes to reduce the fatality rates. To validate our model, a dataset of more than 52k records with 18 informative attributes provided by VAERS has been utilized, and three supervised learning algorithms have been implemented in Python which are Decision Tree, Support Vector Machine, and Naïve Bayes to conduct two experiments. A simple splitting percentage method was performed in the first one, while a k-Folds Cross-validation technique was used in the second experiment with k = 5. The model showed a promising result with stable performance in both experiments, the Decision Tree outperformed other algorithms with a predictive rate of 0.91999 in the first experiment, and 0.91369 in the second one.

Keywords: Covid-19, Vaccine, Adverse reactions, Machine learning (ML), Pfizer/BioNTech, Moderna, Johnson & Johnson

1. Introduction

In December 2020, Food and Drug Administration (FDA) issued Emergency Use Authorizations (EUAs) for the Pfizer/BioNTech and Moderna vaccines for the prevention of COVID-19. In February 2021, Johnson & Johnson's Janssen COVID-19 vaccine became the third vaccine available under EUA [1]. Within a year of the first vaccines being authorized, over 101 million individuals in the United States (US) had been fully vaccinated against COVID-19 [2]. Between March 2020, and January 2021, COVID-19 vaccinations in the US, included approximately 378,039 deaths, and 1.38 million hospitalized cases [3].

Producing a vaccine in a such short time is unprecedented, and the success of these vaccines went beyond expectations, yet several challenges arise. For instance, the nature of the human body's reactions to these vaccines that diverse from one individual to another. Medical history, chronic disease, and age are other factors affecting the condition of vaccine recipients [4] [5].

Despite the large number of people who have received these vaccines safely, some signs (e.g., Pain, Redness, Fever, Headache, and others) have occurred, which is a normal indicator that their body is building protection. However, adverse reactions have been observed that could cause long-term health problems. These reactions generally

Received 25 June 2022; revised 2 October 2022; accepted 10 October 2022.
Available online 3 May 2023

* Corresponding author.
E-mail address: moh.b.m@tu.edu.iq (M.B. Albayati).

https://doi.org/10.33640/2405-609X.3271

happen within six weeks of receiving a vaccine dose and could cause a severe allergic reaction.

Myocarditis (inflammation of the heart muscle), Pericarditis (inflammation of the lining outside the heart) are some of these side effects. Reporting rates for death events increased with increasing age, and males generally had higher reporting rates than females. According to the Vaccine Adverse Event Reporting System (VAERS), reactions reported after getting a booster shot are similar to those after the two-dose or single-dose primary shots. Fever, fatigue, and pain at the injection site were the most commonly reported side effects, and overall, most side effects were mild to moderate. However, serious side effects are rare but occur.

Technology played a vital role in the healthcare systems and proved its efficiency on many occasions over the years [6] [7].Among many employed technologies, (ML and its application were the most implemented technology. Experts exploited this field (ML) in various areas of healthcare, including chronic diseases like diabetes [8], liver disorder [9], breast cancer [10] [11], and COVID-19 [12].

In this work, we exploited the supervised ML techniques against the most popular vaccines (Pfizer/BioNTech, Moderna, and Jensen Johnson & Johnson) to design a prediction model for predicting possible adverse reactions after getting the vaccine. To this end, three supervised algorithms (e.g., Decision Tree DT, Support Vector Machine SVM, and Naïve Bayes algorithms NB) are implemented in Python with the VAERS dataset [13]. VAERS is an epidemiological database maintained jointly by the CDC and FDA since 1990. The dataset has 18 informative attributes with 52,214 cases, considered from (Jan. 2021–Nov. 2021). The dataset was targeted with a "DIED" class label (9657 positive, and 42,457 negatives).

Preparing VAERS data for this work, we statistically analyze the dataset to discover useful information for supporting the decision-making process. The analysis found that the dataset was massive, narrative, noisy, and contain redundant information. Thus, a series of heavy preprocessing steps are performed to prepare the clean data for ML optimization.

To validate our model, two experiments were conducted, in the first experiment, a simple (30–70) train-test splitting method is performed, where 70% (36,479 records) of the data was selected as training, and the rest 30% (15,635 records) was used as a testing set. It is worth mentioning that this ratio of splitting is recommended as an ideal proportion [14]. The k-Folds Cross-validation technique was performed with $k = 5$ in the second experiment splitting the data into 20–80 ratios for each round. We considered $k = 5$ because experimentally we found that it is sufficient with large samples and assure fair and unbiased class distribution in the dataset [15]. Moreover, a large k value leads to less variance across the training set and limits the model performance across the experiment rounds [16].

Results showed that DT outperformed other algorithms in both experiments with an accuracy of 0.91999 and 0.91369. (See section 5 for more details).

We believe the research presented in this paper is a promising approach, in which ML tools have been successful incorporation to predict the severity of post-vaccination side effects to make a better prediction and move towards better healthcare services. Moreover, The current study adds to a literature that has yielded mixed results with respect to the adverse reactions of the COVID-19 vaccine, and numerous contributions through this work are considered strengths that justify its importance and appropriateness in terms of linking it with the previous researches which conceptualize the total of the work's substantive values including:

- A massive sample size was used (≈50,000) records.
- The death cases considered in the analyzing process of this model; approximately 10.000 instances, representing the severe signs and associated information with the vaccine that escalated to death.
- A promising ML model that has been dedicatedly developed for this purpose with high predicting rates.
- The diverse data types utilized in our model with categorical, nominal, and numerical, as well as a diverse category of each attribute (up to 40 categories).

The remains of this paper are organized as follows: Section 2 reviews the literature for related works along with background materials. Section 3 discusses the research methodology. The model's framework including the dataset and model's implementation is demonstrated in section 4. In section 5 the obtained results are discussed. Finally, the conclusion of this work is presented in section 6.

2. Background materials and related works

Machine Learning is a branch of artificial intelligence (AI) that exploits computational algorithms and enables computers\machines to simulate cognitive behavior to find a trend or patterns (Unsupervised), or to learn from experience to make a decision or give a prediction (Supervised).

Nowadays, ML penetrates plenty of industries, building up real-life intelligent applications, like GPS systems, search engines, health care services, and many others [17] [18]. ML has been adopted in different approaches in healthcare. In this work three supervised algorithms are employed, which are:

- The decision Tree (DT) algorithm: is a supervised technique that generates classification rules by breaking down the dataset into smaller and smaller subsets, forming a tree until the decision node (class label) is met. Each node in the tree represents an attribute of the training set, the leaves hold the class label while the root represents the attribute with the highest information gain. DT is a recursive algorithm that calculates the attributes' information gain in each iteration and selects the most dominant attribute with higher information gain as a splitting criterion, Thereafter, entropy and gain scores would be calculated again among the other attributes. Thus, the next most dominant attribute is found. This process is repeated (recursive), forming a tree, until a decision is reached. The information gain of a specific attribute is calculated by subtracting the entropy of that attribute from the entropy of the dataset, [18] [19]. Calculated in equation (1):

$$\mathbf{Gain}(\mathbf{A}) = \mathbf{Info}(\mathbf{D}) - \boldsymbol{Info}_{A}(\mathbf{D}) \tag{1}$$

where:

Gain (A): Information gain for attribute A.

D: Dataset.

Info (D): (Entropy of D) expected information needed to classify a tuple in D, calculated as follows:

$Info(D) = -\sum_{i=1}^{m} Pi \log_2(Pi)$

m = *Distinct number of the classes in D*

$i = 1,2, \ldots, m.$

$C_i$nbsp;= *Classes in D* ($C_1, C_2, \ldots, C_m$).

$P_i$nbsp;= *Probability that tuple in D belongs to class* C_i

$Info_A(D)$: *(Entropy of A) expected information needed to classify a tuple in D if the tuples are partitioned according to the attribute's values of A.*

$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$

j = *Number of the partitions in D by the attribute A.*

$\frac{|D_j|}{|D|}$nbsp;= *jth weight for splitting the tree.*

- Support Vector Machine (SVM) algorithm is a classification technique designed to define a hyperplane that classifies the training data, it searches for the hyperplane with the widest margin to separate the data classes. The data points that are closest to the hyperplane are called support vectors. SVM calculates the distance between the given object and the hyperplane that separates the class labels [20].
- Naïve Bayes (NB) is a simple probabilistic classifier, based on applying a conditional probability with the independent assumptions between the attributes. This algorithm is easy to build, with no complicated iterative parameter which makes it particularly useful for a huge dataset. A Naïve Bayesian classifier works on the concept that is, each attribute has its effect on a given class, regardless of any correlations to other attributes in the classification process; in this sense is considered a "Naïve".

Bayes rules adopted in this algorithm stated a conditional probability of a certain event based on previous knowledge about that event [19] [20]. As the mathematical equation (2) demonstrates:

$$\boldsymbol{P}(\boldsymbol{C}|\boldsymbol{X}) = \frac{\boldsymbol{P}(\boldsymbol{X}|\boldsymbol{C})\,\boldsymbol{P}(\boldsymbol{C})}{\boldsymbol{P}(\boldsymbol{X})} \tag{2}$$

where:

P(C|X): Posterior probability of class given predictor.

P(X|C): Likelihood which is the probability of predictor given class.

P(C): Prior probability of class.

P(X): Prior probability of predictor.

Reviewing the literature, many works and studies were conducted focusing on the phenomena of COVID-19, the search scope was massive, and researchers in this field deal with it differently according to their perspectives. Our standpoint in this work was deploying a machine learning approach for the COVID-19 vaccine's validity. In this regard, works with a similar approach are presented in the literature. To begin with, Liu et al. [21] proposed a ML approach with a supervised model for determining the survival status of the infected cases to distinguish the patients who require immediate medical assistance, allowing them to have a high priority and reduce the risk rates. To this end, the authors assessed the critical biomarkers of these patients using XGboost algorithms with a set of attributes that include on-set symptoms like (fever, cough, chills, pain, and others), lab \ blood test indicators like (lactate dehydrogenase, and white blood cell count, urea, glucose, and others). Using a dataset of 404 sample cases (213 recovered, and 191 died). The experimental results showed a more than 90% predicting rate.

Following the same vein, Sujath et al. [22] proposed a prediction model using a dataset from the

Kaggle website related to the pandemic in India. The authors developed their model using linear regression, multilayer perceptron, and vector autoregression and employed a correlation method to find dependencies among the epidemiological attributes in the dataset to predict the confirmed death and recovered cases on the daily basis.

Another ML-based system was proposed by Rehman et al. [23], where a prediction system for the COVID19 pandemic has presented, using an x-ray image of the infected patients. The authors employed five different classifiers (Decision Tree, Naïve Bayes, K-NN, Random Forrest, And Support Vector Machine), in addition, an ensemble technique is performed, which is a combining method for applying multiple algorithms at the same time, exploiting a set of symptoms including (diarrhea, voice type, smelling issues, joint pain, dry cough, vomiting, breathing problems, and others), and for the model evaluation purposes, hard voting and soft voting along with k-fold analysis are performed, the experiments showed that the model scored 97% predictive rates, claiming that this percentage has surpassed detection rates of other related works at that time.

On the other hand, Tiwari et al. [24] proposed a time series forecasting model to predict the numbers of confirmed, recovered, and death cases in India. The authors developed the model using patterns taken from China, utilizing ML methods to predict the future forecast spread of COVID-19 based on present scenarios, giving a firm recommendation about the peak of the pandemic across India, along with numbers, dates, and statistics.

Punn et al. [25] presented ML and deep learning techniques for globally epidemic analysis, intending to explain the virus behavior of everyday exponential spreading. They utilized real-time information about the virus across the world and implemented several ML algorithms in this work like support vector regression (SVR), polynomial regression (PR), and several deep learning models. Results showed that the possible number of cases around the world in the next 10 days (at the time of the research) can be predicted.

A visual approach to the ML diagnosis model is proposed by Abd Elaziz et al. [26], where the cases are classified into COVID and non-COVID patients based on x-ray images. The diagnosis process in this work starts with 'feature extraction' to select the important descriptors from the x-ray images, and due to the large size of the images, a parallel implementation architecture is utilized to enhance and accelerate the extraction process of the necessary features. Finally, mathematical modeling is utilized for selecting the most important features. These features are fed to the diagnosis model, and the diagnosis rate achieved by the system was 96.09% and 98.09% over two datasets collected from multiple sources.

Muhammad et al. [27] developed a COVID-19 data mining model using supervised algorithms to predict infected patients' recovery using a dataset from the Kaggle website regarding South Korea. Six algorithms were utilized in this study: Decision Tree DT, Support Vector Machine SVM, Naïve Bayes NB, K Nearest Neighbor K-NN, Logistic Regression LR, and Random Forest RF. Five attributes were used which are Gender, Age, Infection_case, No_day, and State. The experiment results showed high predication rates with 99.85 for DT to 97.49 for LR.

An interesting approach to finding the correlation between COVID-19 and the weather is presented by Fadli et al. [28], where the authors exploited weather elements: average temperature, average humidity, and the average duration of sunlight from June to July of 2020 in Surabaya/Indonesia, using the ID3 decision tree of the data mining algorithms. The study finds a significant correlation between these elements and the number of COVID-19 confirmed cases, and the ID3 had high-performance accuracy of 96.77%.

Batista et al. [29] proposed an ML model for predicting and assessing the risk of positive COVID-19 cases for prioritizing patients' health care assistance. The proposed model consists of a dataset of 235 records, including 102 confirmed COVID-19 cases, collected from a general hospital in São Paulo/Brazil, from 17th to 30th of March 2020, the dataset consists of a set of laboratory attributes: Hemoglobin, Platelets Red blood cells, Leukocytes, Lymphocytes, Monocytes, Basophils, Eosinophils, reactive protein, age, and sex attributes.

To validate the proposed model, five ML algorithms have been utilized which are, Neural Networks NN, Random Forests RF, Gradient Boosting Trees GRT, Logistic Regression LR, And Support Vector Machines SVM, the experimental results indicate that SVM outperformed the other algorithms. In the same context, Hatmal et al. [30] presented a cross-sectional study in Jordan using ML techniques for assessing the post-vaccination side effects. An online survey was developed for this purpose and circulated via social media platforms like WhatsApp, Facebook, and Instagram, from the 9th to the 15th of April 2021. The survey presented as a questionnaire consists of 58 (yes or no) questions, regarding the demographic, medical records, and post-vaccination signs of participants. 2237 respondents were the sample size of this study. To develop the predictor; many ML tools are used, that is Multilayer Perceptron (MLP), Extreme Gradient

Boosting (XGboostq), Random Forest (RF), and K-Star (K*).

Avasarala et al. [31] present an analysis study regarding the incidence of new-onset seizures following COVID-19 vaccines and compared it with the seizures associated with influenza vaccines as a reference, by employing several demographic characteristics and injection site reactions of the individuals who took these vaccines. The study finds that seizures associated with COVID-19 were 3.191, and 0.090 for influenza vaccines claiming that most of the seizures occurred within the first 2 days of the vaccination, and to expand the work another two commonly reported side effects; headache and injection−site reactions like pain, rash, and swelling are included in this study. Other indicators revealed by this study showed that incidences of 92.1 and 52.3 times higher for COVID-19 vaccines compared to influenza vaccines for headache and injection site reactions, respectively. Saad et al. [32] study the adverse events followed by the second dosage of the COVID-19 vaccine to predict three events which are: not survived, recovered, and not recovered by using a total of 4351 instances from the VAERS dataset, using three attributes: 'RECOVERED', 'DIED', and 'SYMPTOM_TEXT' for multiclass classification, and two attributes: 'DIED' and 'SYMPTOM_TEXT' for the binary classification. Three experiments have been running, which are: term frequency-inverse document frequency TF-IDF, a bag of words BoW, and global vectors GloVe.

Several machine learning algorithms including Random Forrest, Ada Boost, Logistic regression., Multilayer perceptron, Gradient Boosting Machine, k-Nearest Neighbor, Stochastic Gradient Descent Classifier, and Extra Tree Classifier were applied, using a dataset of size 5351 records. In addition, the used dataset was balanced by applying two techniques for better prediction rates, known as the synthetic minority oversampling technique (SMOTE) and adaptive synthetic (ADASYN). The results showed that the model gained significant accuracy rates after applying the data balancing techniques.

Another study aimed to discover possible common causes for post vaccines side effects to predict them proposed by Ahamad et al. [33], the authors consider the participant's medical history and look into the post-vaccination adverse reactions of 5209 cases. Statistical analysis is applied to search the similar characteristics that were significantly associated with poor participants' reactions in the majority of the cases.

The patient medical history is strongly related to the vaccine's side effects, according to the findings of this study, where some of these signs are associated with severe disease and even death. Furthermore, a set of significant side effects developed as post-vaccination symptoms, and these indicators need to be taken into account, like age, gender, allergic history, and.

Some other signs found as a poor outcome for the vaccinated patients, like pyrexia, headache, dyspnea, chills, fatigue, various kind of pain, and dizziness.

ML tools were employed in this work by utilizing the aforementioned signs to check the patients with the most likely of having vaccine complications, with an accuracy score above 85%. By Following the same steps, Sujatha et al. [34] utilized various machine learning algorithms, which are Logistic Regression LR, Adaboost AD, Decision Tree DT, and Random Forest RR to develop a prediction model. They considered the DIED variable as the target variable, with a dataset collected from healthcare workers, government bodies, and medical research organizations.

Regarding the experimental results, Adaboost algorithm showed appreciable results with 98.1%, followed by Random Forest with 97.8%, then Logistic Regression with 97.31%, while Decision Tree registered 97.3% predictive rates.

Briefly, we observe many works employing ML/Data mining techniques for detecting COVID-19, adapting various approaches like Hatmal et al. [30] and Ahamad et al. [33]. However, other studies utilized ML for detecting COVID and diagnoses as in [21-29]. On the other, hand researchers like Alhazmi et al. [35] present a statistical study to evaluate the side effects associated with COVID-19 vaccines in Saudi Arabia. On the same page, El-Shitany et al. [36] assess the adverse reactions of Pfizer/BioNTech in the retrospective cross-sectional study. While Chapin-Bardales et al. [37] analyzed COVID-19 post-vaccinations side effects in a statistical study.

3. Research methodology

This study evaluates the COVID-19 vaccine's adverse reactions for Pfizer/BioNTech, Moderna, and J&J in the post-COVID Era. To this end, supervised machine learning algorithms (Decision Tree, Support Vector Machine, and Naïve Bayes algorithms) are utilized to develop a prediction model. The model is implemented in Python using a dataset of 52,114 cases, considered from (Jan 2021 to Nov 2021), with 18 informative attributes associated with a vaccine as side effects. The dataset targeted with "DIED" class labels (positive and negative) 9657 and 42,457 respectively.

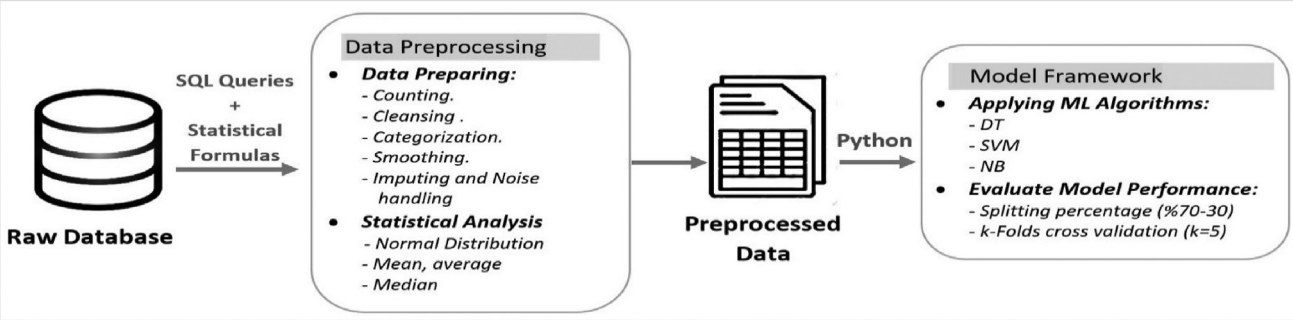

Fig. 1. Research methodology.

To make sure that it is appropriate to the ML techniques, we statistically analyze the dataset to discover useful information for supporting the decision-making process. The analysis found that the dataset was massive, narrative, noisy, and contain redundant information. Thus, a series of heavy preprocessing steps are performed to prepare data for ML optimization shown in Fig. 1. To validate the proposed model, two experiments were conducted: in the first experiment, a simple splitting percentage method was performed with 70–30% splitting ratio, and k-Folds Cross-validation technique was used in the second experiment, as it designed to run for five rounds of testing (k = 5).

4. Research framework

The research model utilizes 52,114 records including different types of attributes based on a set of medicinal, laboratory, and epidemiological features.

4.1. Dataset description

The dataset that has been utilized in this study is adopted from the Vaccine Adverse Event Reporting System (VAERS). It is a passive warning system that aims to detect possible safety problems in U.S.-licensed vaccines. To provide valuable information, both healthcare professionals and vaccine manufacturers report all unusual or unexpected patterns of adverse events. VAERS then accepts and analyzes reports of adverse events (possible side effects).

VAERS data is accessible by downloading raw data in (CSV) form. De-identified VAERS data are available 4–6 weeks after the report is received. VAERS data change as new reports are received, so results may change if the same search is done at a later date. The dataset consists of 52,114 records including 2 numerical, 8 nominals, and 7 categorical attributes, storing clinical, medicinal, laboratory, and epidemiological information. The dataset has also class labels that are distributed as 20% Positive, and 80% negative cases. Below, a brief description of these attributes is given:

1. VAERS_ID: A sequence of numbers used as an identification.
2. SYMPTOM1_CAT: Adverse Event MedDRA Term1, grouped into 41 categories (Abdominal Symptoms, Anxiety, Cough, Chills, …).
3. SYMPTOM2_CAT: Adverse Event MedDRA Term 2 is grouped into 35 categories (Asthenia, Back Pain, Chest discomfort, Decreased appetite, …).
4. SYMPTOM3_CAT: Adverse Event MedDRA Term 3 is grouped into 27 categories (Lymphadenopathy, Pain in extremity, Myalgia, Palpitations, …).
5. SYMPTOM4_CAT: Adverse Event MedDRA Term 4 is grouped into 22 categories (Blood Symptoms, Vomiting, Urticaria, Covid-19, …).
6. SYMPTOM5_CAT: Adverse Event MedDRA Term 5. grouped into 13 categories (Fatigue, Body Pain., Skin Swelling, Rash Symptoms, Vomiting, …).

   (SYMPTOM 1–5) The data in these fields are equivalent to the PT TERM from the MedDRA codebook. MedDRA terms are extracted from the narrative text in VAERS 2 (Item 18–19, See[1] for more details).

7. Age in Years (AGE_YRS_INT): Vaccine recipient's age is categorized in a range of decades (20–30), (30–40), …, (+80).
8. SEX: Sex of the vaccine recipient (Male, Female, Unknown ⧹ Missing).
9. Life-Threatening (L_THREAT): If the vaccine recipient had a life-threatening event associated with the vaccination a "Yes" is placed is used; otherwise the field will be "No".

[1] https://vaers.hhs.gov/docs/VAERSDataUseGuide_en_September2021.pdf.

10. Hospitalized (HOSPITAL): If the vaccine recipient was hospitalized as a result of the vaccination a "Yes" is used; otherwise the field will be "No".
11. Disability (DISABLE): If the vaccine recipient was disabled as a result of the vaccination a "Yes" is placed in this field; otherwise the field will be "No".
12. Other Medications (OTHER_MEDS_CAT): this attribute identifies if the recipient at the time of vaccination is on any type of drug.
13. The healthcare office (OFC_VISIT): Doctor or other healthcare provider office/clinic visit at the time of vaccination.
14. Emergency Room Visit (ER_ED_VISIT): An emergency room or urgent care at vaccination time.
15. Vaccination doses VAX_DOSE_SERIES_CAT: Number of doses administered (1, 2, +2, Unknown \ Missing).
16. DIED: this field represents the class label of the dataset, If the vaccine recipient died a "Positive/Yes" is used; otherwise, the field will be "Negative/No".
17. ALLERGIES_CAT: this attribute determines the pre-existing and any type of allergies that existed at the time of vaccination.
18. Current Illnesses (CUR_ILL): contains a narrative about any illnesses at the time of the vaccination.

Some other attributes were ignored, as they had no informative role in the prediction process, like VAX_LOT, VAX_SITE, VAX_NAME, RECVDATE, STATE, and alike.

4.2. Dataset statistical analysis

Tables 1–10 show in numbers the statistics and distribution of the dataset sample. Some attributes (categorical) have a continuous; rather than the normal distribution. So, it makes no sense to examine them.

Statistical analysis is one of the core components of the ML, which identifies common patterns and trends in the data samples, to understand the data analysis, and how the results of the developed model were acquired [20].

4.3. Data preprocessing

To improve data quality, the data is preprocessed to transform raw data into a structural form that applies to the ML techniques, through cleaning and removing unwanted data hence obtaining more accurate results [38].

Table 1. Age distribution.

Age_YRS	Count	%
10–19	361	0.006927
20–29	5817	0.111621
30–39	10,667	0.204686
40–49	10,049	0.192827
50–59	8827	0.169379
60–69	6057	0.116226
70–80	3671	0.070442
80+	4352	0.083509
Unknown \ Missing	2313	0.044383
Total	52,114	1
Average Age	50.027	

Table 2. Label distribution.

Class Label	Count	%
Yes \ Positive	9657	0.185305
No \ Negative	42,457	0.814695
Total	52,114	1

Table 3. Life threat distribution.

L_Threat	Count	%
Yes \ Positive	932	0.017884
No \ Negative	51,182	0.982116
Total	52,114	1

Table 4. Doctor office/clinic visit distribution.

OFC_Visit	Count	%
Yes \ Positive	9237	0.177246
No \ Negative	42,877	0.822754
Total	52,114	1

Table 5. Disability distribution.

Disable	Count	%
Yes \ Positive	303	0.005814
No \ Negative	51,811	0.994186
Total	52,114	1

Table 6. Sex distribution.

Sex	Count	%
Female	38,503	0.738823
Male	12,639	0.242526
Unknown \ Missing	972	0.018651
Total	52,114	1

Table 7. Vaccine dose distribution.

Vax_Dose_Series	Count	%
1	33,679	0.646256
2	9669	0.185536
2+	117	0.002245
Unknown \ Missing	8649	0.165963
Total	52,114	1

Table 8. Vaccine manufacture distribution.

Vacc_Manu.	Count	%
J&J	847	0.016253
Moderna	26,209	0.502917
Pfizer＼BioNtech	24,971	0.479161
Unknown	87	0.001669
Total	52,114	1

Table 9. Hospitalization distribution.

Hospitalize	Count	%
Yes＼Positive	4471	0.085793
No＼Negative	47,643	0.914207
Total	52,114	1

Table 10. ER visit distribution.

ER_ED_Vist	Count	%
Yes＼Positive	9077	0.174176
No＼Negative	43,037	0.825824
Total	52,114	1

The dataset was massive, narrative, noisy, and redundant. First, we had to balance it by choosing the right proportion, depending on class label distribution, as it makes no sense of having data with a 50 to 1 splitting ratio (500,000 records with 10,000 negatives, and 490,000 positives). We had to randomly choose a sample dataset of size (≈52,000) records, with a distribution ratio of (80-20%) for the "Negative", and "Positive" cases respectively. Then, a series of preprocessing steps were performed on each attribute in the dataset as shown in Fig. 2 and listed below:

1. Symptoms counting: count the symptoms' or signs' reputation to determine the highest frequency of symptoms.
2. Initial reviewing: applying basic reviewing to the high frequencies' signs and giving them possible category name (category name 1), plus give a (category name 2, 3, … , N) to the similar signs in the lower frequencies by searching them, to unify them in step 4.
3. Data cleaning and reduction: search the records that contain narrative, irrelevancy, or a big chunk of data. Reduce this amount of information by using uniform phrases and keywords. At this step, misspell checking is also performed, as some symptoms were incorrectly spelled, to recount them with their actual categories in step 2, or to be reconsidered as new categories.
4. Categorization: unify the similar category names, and group the symptoms according to these names.
5. Next, a special label "*Other_Symptoms*" is given to the cases that have less than 50 counts, or individual cases with a special symptom that doesn't belong to a specific category.
6. Data transformation: smoothing the data by discretizing nominal and numerical types into clusters, like the age attribute, which transformed to intervals of decades (20–30, 30–40, … , +80)
7. Noise handling: impute missing values by using the "Unknown＼Missing." label, and eliminates rubbish, incomplete, and contradictory records.

4.4. Model implementation

The model is implemented using Python programming language. Python is an open-source, powerful, flexible, programing language that has a pretty straightforward syntax, giving it the ability to execute complex tasks easily and simply by an emphasis on natural language nature. Python is considered one of the best choices for ML and Artificial Intelligence (AI) projects, due to the tons of libraries and frameworks that significantly cut down on the work required to implement deep neural networks and machine learning algorithms [39].

For the coding environment, we use the Jupyter[2] notebook, which is a web-based interactive environment that supports dozens of programming languages, including Python. For installing the Jupyter notebook we used a distribution platform called Anaconda.[3]

Finally, we develop our SQL queries and statistical formulas for data preprocessing purposes. Fig. 3 gives a sneak peek at the coding environment and the developing libraries.

5. Results and discussion

In this section, the model performance along with the confusion matrix and metrics are presented.

5.1. Evaluation measures

We used some of the most common metrics to evaluate the model performance. However, to understand these measures clearly, we need to shed a light on some terms as follows:

- True Positive TP: the positive cases that were correctly predicted as positive.

[2] https://jupyter.org/about.
[3] https://www.anaconda.com/about-us.

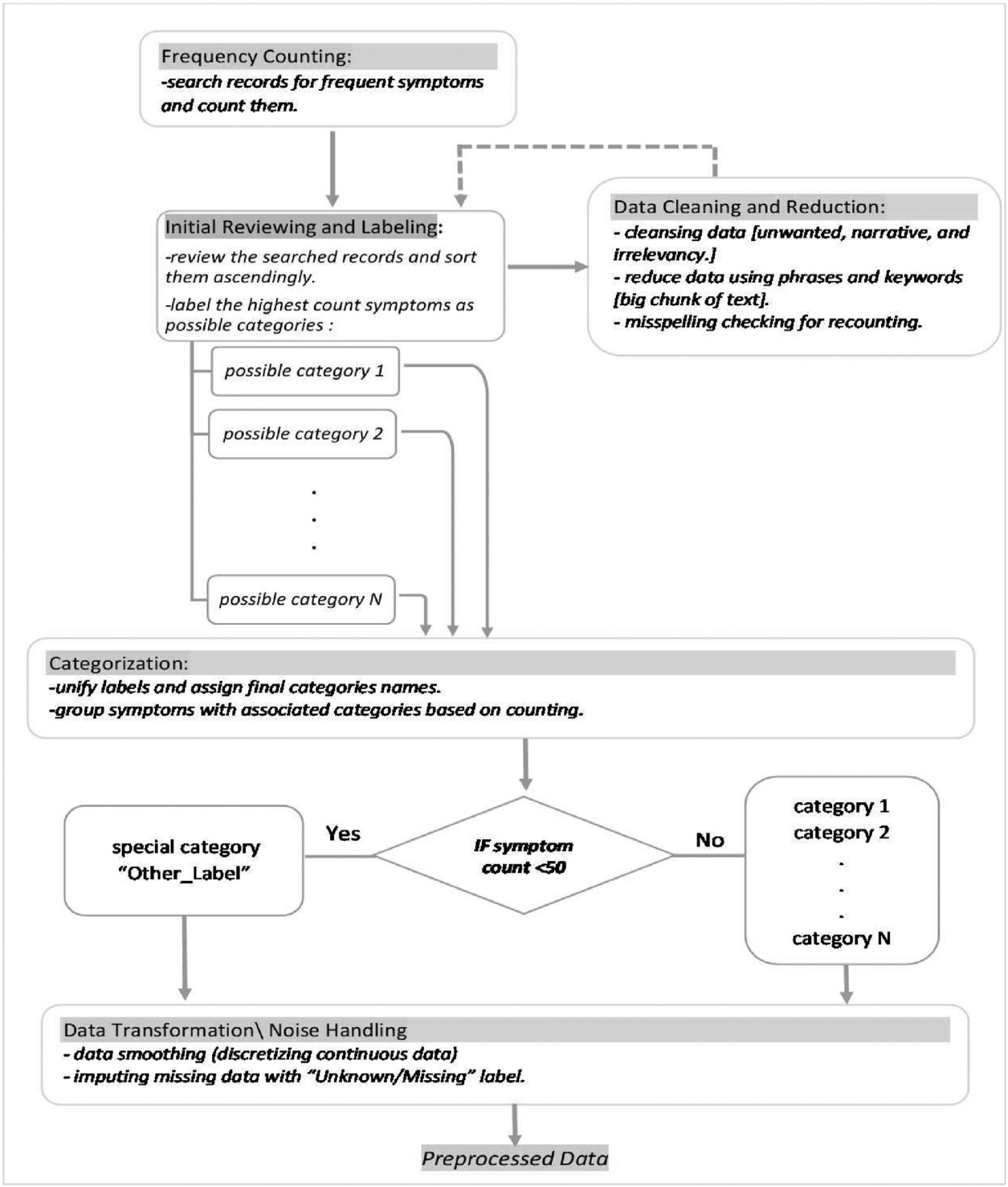

Fig. 2. Preprocessing steps.

- False Positive FP: the negative cases that were incorrectly predicted as positive.
- True Negative TN: the negative cases that were correctly predicted as negative.
- False Negative FN: the positive cases that were incorrectly predicted as negative.

Below are the metrics used for the model evaluation, stated in equations (3)–(7) [40]:

1. Accuracy: fraction of the predicted class that was correct.

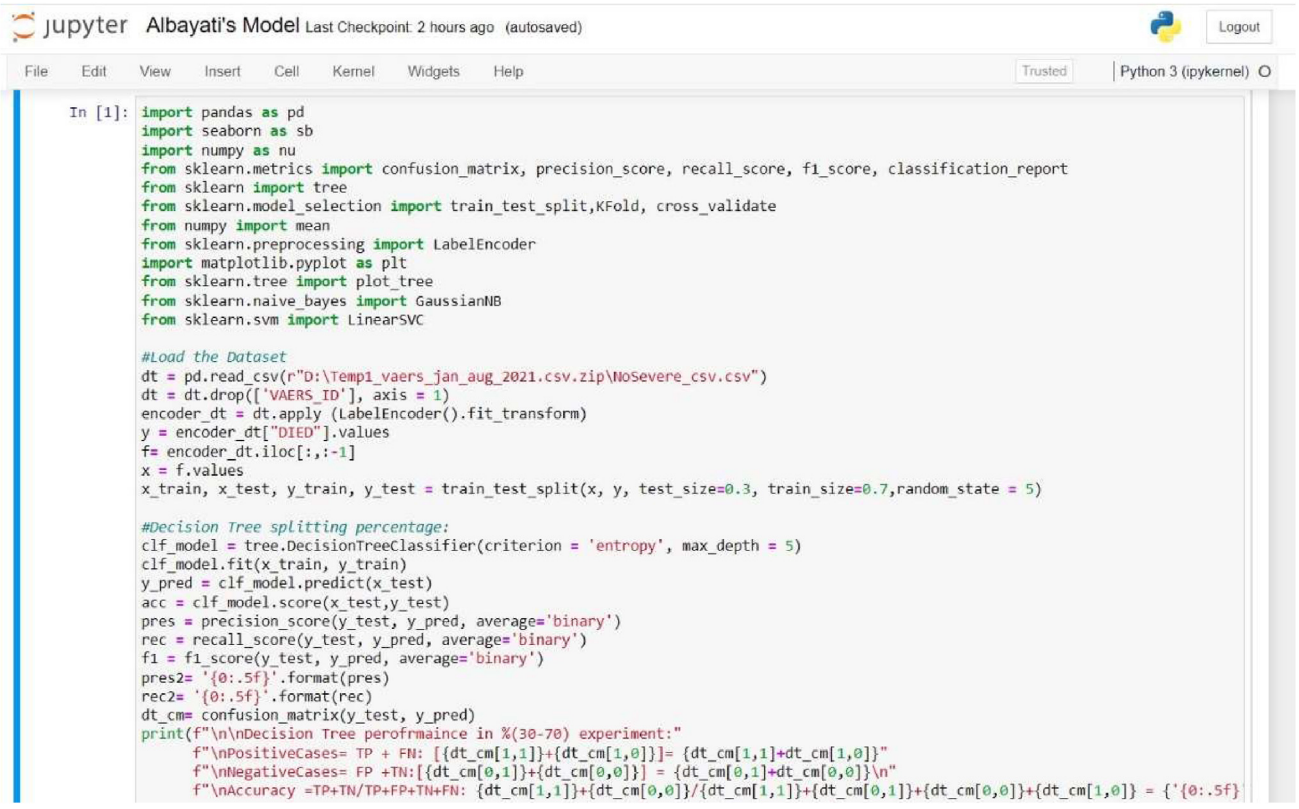

Fig. 3. Coding environment.

$$Acc. = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

2. Precision: fraction of the positive prediction that was correctly identified as positive in the positive predictions' samples.

$$Prec. = \frac{TP}{TP + FP} \quad (4)$$

3. Recall: true positive rate (total number of the positive cases that are correctly identified by the model).

$$Rec. = \frac{TP}{TP + FN} \quad (5)$$

4.Specificity: true negative rate (total number of the negative cases that are correctly identified by the model).

$$Spec. = \frac{TN}{TN + FP} \quad (6)$$

5. F1-score: determine the harmonic mean of the model's precision and recall.

$$F1.Sc. = 2^{*}\frac{Precision^{*}\ Recall}{Precision + Recall} \quad (7)$$

5.2. Experimental results

Two experiments have been conducted to evaluate the proposed model. In the first experiment, a simple splitting method is performed, where 70% of the dataset (36,479 records) was selected as training, and the rest 30% (15,635 records) was used as a testing set. It is worth mentioning that this ratio of splitting is recommended as an ideal proportion [19]. In this experiment, DT outperformed other algorithms with 0.91999 predictive rates, followed by SVM scoring accuracy of 0.86517, and finally, NB registered 0.83908. Next, are the obtained results of the employed algorithms.

>> **Decision Tree Performance in %(30−70 ...**

Positive Cases = TP + FN: [2026 + 906] = 2932

Negative Cases = FP + TN:[345 + 12,358] = 12,703

Accuracy = TP + TN/TP + FP + TN + FN = 0.91999

Precision = TP/(TP + FP) = 0.85449

Recall = TP/(TP + FN) = 0.69100
Specificity = TN/(TN + FP) = 0.97284
F1-Score = 2*(Prec.*Rec.)/(Prec.+Rec.)
nbsp;= 0.76410
>> **SVM Performance in %(30–70) ...**
Positive Cases = TP + FN: [1414 + 1518] = 2932
Negative Cases = FP + TN:[590 + 12,113] = 12,703
Accuracy = TP + TN/TP + FP + TN + FN = 0.86517
Precision = TP/(TP + FP) = 0.70559
Recall = TP/(TP + FN) = 0.48226
Specificity = TN/(TN + FP) = 0.95355
F1-Score = 2*(Prec.*Rec.)/(Prec. + Rec.) =nbsp; 0.57293
>> **Naïve Bayes Performance in %(30–70) ...**
Positive Cases = TP + FN:[2459 + 473] = 2932
Negative Cases = FP + TN:[2043 + 10,660] = 12,703
Accuracy = TP + TN/TP + FP + TN + FN = 0.83908
Precision = TP/(TP + FP) = 0.54620
Recall = TP/(TP + FN) = 0.83868
Specificity = TN/(TN + FP) = 0.83917
F1-Score = 2*(Prec.*Rec.)/(Prec. + Rec.)
nbsp;= 0.66156

Other metrics and indicators are stated in Table 11, and Fig. 4 shows the ROC curves of the employed algorithms.

To test the model's robustness, we needed to examine the performance across the whole dataset rather than 30% only. So, we use the k-Fold cross-validation, which is one of the most common validation techniques to assess the analyzing process, and how it can be generalized over the whole data samples. k-Fold is an iterative technique, that involves partitioning the data into smaller subsets, in each iteration (round), a proportion (fold) of data is selected as test cases, while the remains (k-1) subsets are used for training purposes, this process goes k-times, accordingly with the number of folds, then jump to the next fold for testing, and so on until all folds (whole dataset) are tested [19] [20].

In our experiment, we use 5 folds, to assure fair and unbiased class distribution in the dataset, which randomly splits it into 5 * [1931 + 8492] positive, and negative cases respectively, as Fig. 5 shows, in each round the metrics of Accuracy, Precision, Recall, and F-Score are calculated; plus the average of all rounds is taken at the end of the experiment.

Table 11. Experiment 1 results.

Algorithm	Accuracy	Precision	Recall	F1	Specificity
DT	0.91999	0.85449	0.69100	0.76410	0.97284
NB	0.83908	0.54620	0.83868	0.66156	0.83917
SVM	0.86517	0.70559	0.48226	0.57293	0.95355

In the second experiment, the model exhibited a stable performance, scoring a close result to the (splitting percentage) experiment. Before diving deeper, we need to clarify a few notations that appeared in this experiment; to make these results easier to grasp.

The notations mainly appear in three groups as Fig. 6 shows:

- Group (1): indicates the confusion matrix.
- Group (2): represents the validation metrics.
- Group (3): this block appears at the end of the final round to report the overall performance.

And the gray shaded symbols imply the following:

- Fold N: state the round's number.
- TP_N, FP_N, TN_N, FN_N: hold the numbers of the true\false - positive\negative cases.
- PC_N, NC_N: these notations refer to the total numbers of positive and negative cases in each round.
- Prec., Rec., Acc., F1-Sc., and Spec.: represent the calculated values of (Precision, Recall, Accuracy, F1-Score, and Specificity) respectively.

The next results represent the model's output for the k- Fold Cross-Validation:

Decision Tree performance in k-Fold cross-validation:

Fold 1:
Positive Cases = TP + FN:[934 + 997] = 1931
Negative Cases = FP + TN:[52 + 8440] = 8492
ACCURACY = TP + TN/TP + FP + TN + FN:934 + 8440/8440 + 934+52 + 997 = 0.89936
PRECISION = TP/TP + FP:934/934 + 52 = 0.9473
RECALL = TP/TP + FN:934/934 + 997 = 0.4837
F1 = 2*(Precision * Recall)/(Precision + Recall): 2*(0.9473 * 0.4837)/(0.9473 + 0.4837) = 0.6404
Specificity = TN/(TN + FP):8440/(8440 + 52) = 0.9939

Fold 2:
Positive Cases = TP + FN:[908 + 1023] = 1931
Negative Cases = FP + TN:[146 + 8346] = 8492
ACCURACY = TP + TN/TP + FP + TN + FN:908 + 8346/8346 + 908+146 + 1023 = 0.88784
PRECISION = TP/TP + FP:908/908 + 146 = 0.8615
RECALL = TP/TP + FN:908/908 + 1023 = 0.4702

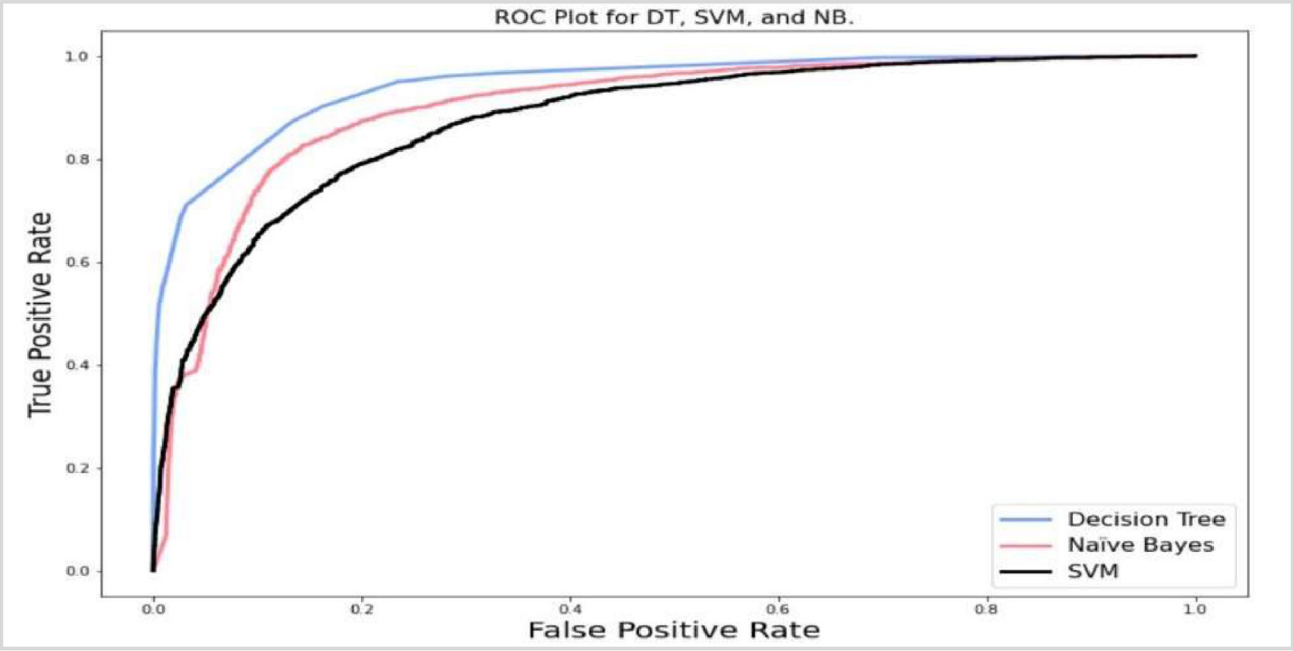

Fig. 4. ROC Curve of the Algorithms.

F1 = 2*(Precision * Recall)/(Precision + Recall): 2*(0.8615 * 0.4702)/(0.8615 + 0.4702) = 0.6084
Specificity = TN/(TN + FP):8346/(8346 + 146) = 0.9828

Fold 3:

Positive Cases = TP + FN:[1363 + 569] = 1932
Negative Cases = FP + TN:[348 + 8143] = 8491
ACCURACY = TP + TN/TP + FP + TN + FN: 1363 + 8143/8143 + 1363+348 + 569 = 0.91202
PRECISION = TP/TP + FP:1363/1363 + 348 = 0.7966
RECALL = TP/TP + FN:1363/1363 + 569 = 0.7055
F1 = 2*(Precision * Recall)/(Precision + Recall): 2*(0.7966 * 0.7055)/(0.7966 + 0.7055) = 0.7483
Specificity = TN/(TN + FP):8143/(8143 + 348) = 0.9590

Fold 4:

Positive Cases = TP + FN:[1297 + 635] = 1932
Negative Cases = FP + TN:[77 + 8414] = 8491
ACCURACY = TP + TN/TP + FP + TN + FN:1297 + 8414/8414 + 1297+77 + 635 = 0.93169
PRECISION = TP/TP + FP:1297/1297 + 77 = 0.9440
RECALL = TP/TP + FN:1297/1297 + 635 = 0.6713

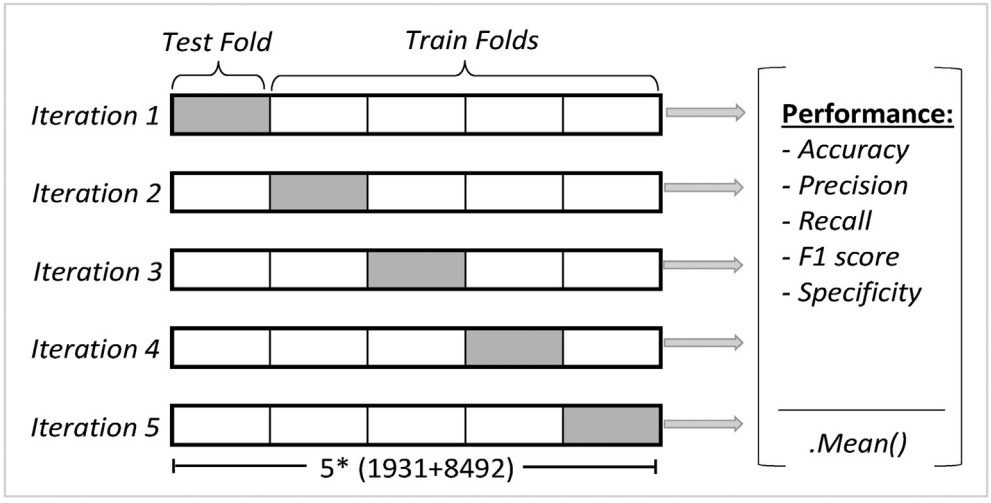

Fig. 5. k-Fold cross-validation experiment.

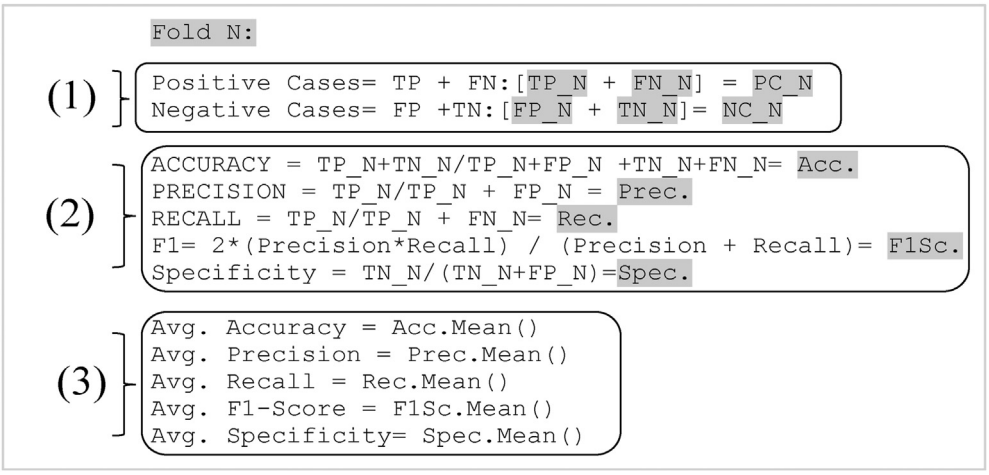

Fig. 6. Experiment 2 notations.

F1 = 2*(Precision * Recall)/(Precision + Recall): 2*(0.9440 * 0.6713)/(0.9440 + 0.6713) = 0.7846

Specificity = TN/(TN + FP):8414/(8414 + 77) = 0.9909

Fold 5:

Positive Cases = TP + FN:[1384 + 547] = 1931

Negative Cases = FP + TN:[104 + 8387] = 8491

ACCURACY = TP + TN/TP + FP + TN + FN:1384 + 8387/8387 + 1384+104 + 547 = 0.93754

PRECISION = TP/TP + FP:1384/1384 + 104 = 0.9301

RECALL = TP/TP + FN:1384/1384 + 547 = 0.7167

F1 = 2*(Precision * Recall)/(Precision + Recall): 2*(0.9301 * 0.7167)/(0.9301 + 0.7167) = 0.8096

Specificity = TN/(TN + FP):8387/(8387 + 104) = 0.9878

Avg. Accuracy = 0.91369

Avg. Precision = 0.89588

Avg. Recall = 0.60949

Avg. F1-Score = 0.71825

Avg. Specificity = 0.98288

Naïve Bayes performance in k-Fold cross validation:

Fold 1:

Positive Cases = TP + FN:[802 + 1129] = 1931

Negative Cases = FP + TN:[0 + 8492] = 8492

ACCURACY = TP + TN/TP + FP + TN + FN:802 + 8492/8492 + 802+0 + 1129 = 0.89168

PRECISION = TP/TP + FP:802/802 + 0 = 1.0000

RECALL = TP/TP + FN:802/802 + 1129 = 0.4153

F1 = 2*(Precision * Recall)/(Precision + Recall): 2*(1.0000 * 0.4153)/(1.0000 + 0.4153) = 0.5869

Specificity = TN/(TN + FP):8492/(8492 + 0) = 1.0000

Fold 2:

Positive Cases = TP + FN:[1647 + 284] = 1931

Negative Cases = FP + TN:[936 + 7556] = 8492

ACCURACY = TP + TN/TP + FP + TN + FN:1647 + 7556/7556 + 1647+936 + 284 = 0.88295

PRECISION = TP/TP + FP:1647/1647 + 936 = 0.6376

RECALL = TP/TP + FN:1647/1647 + 284 = 0.8529

F1 = 2*(Precision * Recall)/(Precision + Recall): 2*(0.6376 * 0.8529)/(0.6376 + 0.8529) = 0.7297

Specificity = TN/(TN + FP):7556/(7556 + 936) = 0.8898

Fold 3:

Positive Cases = TP + FN:[1578 + 354] = 1932

Negative Cases = FP + TN:[1173 + 7318] = 8491

ACCURACY = TP + TN/TP + FP + TN + FN:1578 + 7318/7318 + 1578+1173 + 354 = 0.85350

PRECISION = TP/TP + FP:1578/1578 + 1173 = 0.5736

RECALL = TP/TP + FN:1578/1578 + 354 = 0.8168

F1 = 2*(Precision * Recall)/(Precision + Recall): 2*(0.5736 * 0.8168)/(0.5736 + 0.8168) = 0.6739

Specificity = TN/(TN + FP:7318/(7318 + 1173) = 0.8619

Fold 4:

Positive Cases = TP + FN:[1770 + 162] = 1932.

Negative Cases = FP + TN:[2081 + 6410] = 8491.

ACCURACY = TP + TN/TP + FP + TN + FN:1770 + 6410/6410 + 1770+2081 + 162 = 0.78480.

PRECISION = TP/TP + FP:1770/1770 + 2081 = 0.4596.

RECALL = TP/TP + FN:1770/1770 + 162 = 0.9161.

F1 = 2*(Precision * Recall)/(Precision + Recall): 2*(0.4596 * 0.9161)/(0.4596 + 0.9161) = 0.6121.

Specificity = TN/(TN + FP):6410/(6410 + 2081) = 0.7549.

Fold 5:

Positive Cases = TP + FN:[1789 + 142] = 1931

Negative Cases = FP + TN:[2638 + 5853] = 8491

ACCURACY = TP + TN/TP + FP + TN + FN:1789 + 5853/5853 + 1789+2638 + 142 = 0.73326
PRECISION = TP/TP + FP:1789/1789 + 2638 = 0.4041
RECALL = TP/TP + FN:1789/1789 + 142 = 0.9265
F1 = 2*(Precision * Recall)/(Precision + Recall): 2*(0.4041 * 0.9265)/(0.4041 + 0.9265) = 0.5628
Specificity = TN/(TN + FP:5853/(5853 + 2638) = 0.6893
Avg. Accuracy = 0.82924
Avg. Precision = 0.61499
Avg. Recall = 0.78553
Avg. F1-Score = 0.63309
Avg. Specificity = 0.83917

SVM performance in k-Fold cross-validation:

Fold 1:
Positive Cases = TP + FN:[1475 + 456] = 1931
Negative Cases = FP + TN:[1784 + 6708] = 8492
ACCURACY = TP + TN/TP + FP + TN + FN:1475 + 6708/6708 + 1475+1784 + 456 = 0.78509
PRECISION = TP/TP + FP:1475/1475 + 1784 = 0.4526
RECALL = TP/TP + FN:1475/1475 + 456 = 0.7639
F1 = 2*(Precision * Recall)/(Precision + Recall): 2*(0.4526 * 0.7639)/(0.4526 + 0.7639) = 0.5684
Specificity = TN/(TN + FP:6708/(6708 + 1784) = 0.7899

Fold 2:
Positive Cases = TP + FN:[1151 + 780] = 1931
Negative Cases = FP + TN:[571 + 7921] = 8492
ACCURACY = TP + TN/TP + FP + TN + FN:1151 + 7921/7921 + 1151+571 + 780 = 0.87038
PRECISION = TP/TP + FP:1151/1151 + 571 = 0.6684
RECALL = TP/TP + FN:1151/1151 + 780 = 0.5961
F1 = 2*(Precision * Recall)/(Precision + Recall): 2*(0.6684 * 0.5961)/(0.6684 + 0.5961) = 0.6302
Specificity = TN/(TN + FP):7921/(7921 + 571) = 0.9328

Fold 3:
Positive Cases = TP + FN:[1238 + 694] = 1932
Negative Cases = FP + TN:[449 + 8042] = 8491
ACCURACY = TP + TN/TP + FP + TN + FN:1238 + 8042/8042 + 1238+449 + 694 = 0.89034
PRECISION = TP/TP + FP:1238/1238 + 449 = 0.7338
RECALL = TP/TP + FN:1238/1238 + 694 = 0.6408
F1 = 2*(Precision * Recall)/(Precision + Recall): 2*(0.7338 * 0.6408)/(0.7338 + 0.6408) = 0.6842
Specificity = TN/(TN + FP:8042/(8042 + 449) = 0.9471

Fold 4:
Positive Cases = TP + FN:[494 + 1438] = 1932
Negative Cases = FP + TN:[231 + 8260] = 8491
ACCURACY = TP + TN/TP + FP + TN + FN:494 + 8260/8260 + 494+231 + 1438 = 0.83987
PRECISION = TP/TP + FP:494/494 + 231 = 0.6814
RECALL = TP/TP + FN:494/494 + 1438 = 0.2557
F1 = 2*(Precision * Recall)/(Precision + Recall): 2*(0.6814 * 0.2557)/(0.6814 + 0.2557) = 0.3718
Specificity = TN/(TN + FP):8260/(8260 + 231) = 0.9728

Fold 5:
Positive Cases = TP + FN:[639 + 1292] = 1931
Negative Cases = FP + TN:[416 + 8075] = 8491
ACCURACY = TP + TN/TP + FP + TN + FN:639 + 8075/8075 + 639+416 + 1292 = 0.83612
PRECISION = TP/TP + FP:639/639 + 416 = 0.6057
RECALL = TP/TP + FN:639/639 + 1292 = 0.3309
F1 = 2*(Precision * Recall)/(Precision + Recall): 2*(0.6057 * 0.3309)/(0.6057 + 0.3309) = 0.4280
Specificity = TN/(TN + FP:8075/(8075 + 416) = 0.9510
Avg. Accuracy = 0.84436
Avg. Precision = 0.62838
Avg. Recall = 0.51746
Avg. F1-Score = 0.53652
Avg. Specificity = 0.91872

Here, Table 12 summarizes experiment 2 results.

To understand the capability of the proposed work, we compare it with similar approaches reviewed earlier, although the related works reveal significant findings regarding the COVID-19 vaccination adverse effects. However, a few limitations need further research effort, they have used a comparatively small amount of patient data. Therefore, to make a generalized decision, it is important to have a deep analysis with larger population size, this shortcoming is resolved here by employing a massive data sample with (52,5114) records against 2237, 5351, 5209, and 4417 in [30] [31], [32], [33], and [34] respectively.

Another noticeable point is the absence of critical (death) cases, which is an important aspect of analyzing the COVID-19 vaccine, where [30] [31] [33], and [34] employed only mild-moderate symptoms. However, [32] utilizes this indicator, but with only 3 attributes associated with it. In this work, we employed the class (DIED) with more than 10,000 cases and 18 informative attributes. Adding the diverse types of data, including (categorical, nominal, numerical, and discrete), as well as the diverted categories in each attribute (up to 40 categories).

In contrast, other works only considered limited types of data, which is another distinguished point that differentiates our work from others, to name a

Table 12. Experiment 2 results summary.

Folds		Accuracy	Precision	Recall	F1	Specificity
Decision Tree	1St	0.89936	0.9473	0.4837	0.6404	0.9939
	2nd	0.88784	0.8615	0.4702	0.6084	0.9828
	3rd	0.91202	0.7966	0.7055	0.7483	0.959
	4th	0.93169	0.9440	0.6713	0.7846	0.9909
	5th	0.93754	0.9301	0.7167	0.8096	0.9878
	Avg.	0.91369	0.89588	0.60949	0.71825	0.98288
Naïve Bayes	1St	0.89168	1	0.4153	0.5869	1
	2nd	0.88295	0.6376	0.8529	0.7297	0.8898
	3rd	0.8535	0.5736	0.8168	0.6739	0.8619
	4th	0.7848	0.4596	0.9161	0.6121	0.7549
	5th	0.73326	0.4041	0.9265	0.5628	0.6893
	Avg.	0.82924	0.61499	0.78553	0.63309	0.83917
SVM	1St	0.78509	0.4526	0.7639	0.5684	0.7899
	2nd	0.87038	0.6684	0.5961	0.6302	0.9328
	3rd	0.89034	0.7338	0.6408	0.6842	0.9471
	4th	0.83987	0.6814	0.2557	0.3718	0.9728
	5th	0.83612	0.6057	0.3309	0.4280	0.9510
	Avg.	0.84436	0.62838	0.51746	0.53652	0.91872

few: [31] employ basic post-vaccination signs like fatigue, fever, vaccine site pain … etc., and [32] utilizes 3 attributes considered with patients that had two vaccine doses only, while in our study first, second and even +2 administered doses in some cases were utilized in the dataset.

In [34], a similar VAERS dataset was used, but we observe that these results differ from ours where different approaches and data preprocessing steps were followed, as well as a different algorithm. Finally, the authors of [32] focused mainly on the performance side of the research by employing numerous techniques, like algorithms voting method to improve the classifier accuracy, and data balancing approaches to avoid model overfitting and get better results. While, the main scope of this work was to study human body reactions against the vaccine, by straightforwardly applying ML tools without using any performance enhancement skills.

6. Conclusion

In this paper, the COVID-19 vaccine validity is investigated by studying the human body's reaction. An Informative dataset was adopted from the Vaccine Adverse Event Reporting System (VAERS) with 18 attributes and more than 52k cases to train and test a prediction model based on a supervised learning approach.

A sequence of intensive statistical analysis and preprocessing steps was taken. Then three supervised learning algorithms are implemented in Python (Decision Tree DT, Support Vector Machine SVM, and Naïve Bayes NB) to conduct two experiments.

The experiments aim to evaluate the accuracy of the proposed prediction model; in the first experiment a 30–70 splitting ratio was used, as 30% of the dataset was set for testing, and 70% for training. While in the second one, the whole dataset was tested in the k-Fold cross-validation method, where the model tests the algorithms for 5 rounds, splitting the data into 20–80 ratios for each round.

In both experiments, the model exhibited promising results, along with a steady and robust performance with an accuracy of 0.91999 and 0.91369 for the DT algorithm.

In future work, we are planning to further extend our work to other vaccinations and disease datasets. This can be taken a step further for predictions based on multiple symptoms. Moreover, more algorithms can be considered to develop an automated prediction system. A comparison then can take place to determine the most accurate predicting algorithm.

References

[1] J.E.K. Hildreth, D.J. Alcendor, Targeting covid-19 vaccine hesitancy in minority populations in the us: implications for herd immunity, Vaccines 9 (2021) 489–501, https://doi.org/10.3390/vaccines9050489.

[2] M. Birhane, S. Bressler, G. Chang, T. Clark, L. Dorough, M. Fischer, L.F. Watkins, J.M. Goldstein, K. Kugeler, G. Langley, K. Lecy, S. Martin, F. Medalla, K. Mitruka, L. Nolen, K. Sadigh, R. Spratling, G. Thompson, A. Trujillo, COVID-19 vaccine breakthrough infections reported to CDC-United States, January 1–April 30, 2021, MMWR Morb. Mortal. Wkly. Rep. 70 (2021) 792–793, https://doi.org/10.15585/mmwr.mm7021e3.

[3] S.H. Woolf, D.A. Chapman, R.T. Sabo, E.B. Zimmerman, Excess deaths from COVID-19 and other causes in the US, March 1, 2020, to January 2, 2021, JAMA 325 (2021) 1786–1789, https://doi.org/10.1001/jama.2021.5199.

[4] A. Remmel, COVID vaccines and safety: what the research says, Nature 590 (2021) 538–540, https://doi.org/10.1038/d41586-021-00290-x.

[5] J. Sprent, C. King, COVID-19 vaccine side effects: the positives about feeling bad, Sci. Immunol. 6 (2021) 256–268, https://doi.org/10.1126/sciimmunol.abj9256.

[6] F. Fernandes, H. Vicente, A. Abelha, J. Machado, P. Novais, J. Neves, Artificial neural networks in diabetes control, in: 2015 Sci. Inf. Conf., IEEE, 234, 2015, pp. 362–370, https://doi.org/10.1109/SAI.2015.7237169.

[7] J.A. Cruz, D.S. Wishart, Applications of machine learning in cancer prediction and prognosis, Cancer Inf. 2 (2006) 22–41, https://doi.org/10.1177/117693510600200030, 117693510600200030.

[8] S.I. Ayon, M.M. Islam, Diabetes prediction: a deep learning approach, Int. J. Inf. Eng. Electron. Bus. 12 (2019) 21–27, https://doi.org/10.5815/ijieeb.2019.02.03.

[9] M.R. Haque, M.M. Islam, H. Iqbal, M.S. Reza, M.K. Hasan, Performance evaluation of random forests and artificial neural networks for the classification of liver disorder, in: 2018 Int. Conf. Comput. Commun. Chem. Mater. Electron. Eng., IEEE, 887, 2018, pp. 1–5, https://doi.org/10.1109/IC4ME2.2018.8465658.

[10] M.M. Islam, H. Iqbal, M.R. Haque, M.K. Hasan, Prediction of breast cancer using support vector machine and K-Nearest neighbors, in: 2017 IEEE Reg. 10 Humanit. Technol. Conf., IEEE, 32017, 2017, pp. 226–229, https://doi.org/10.1109/R10-HTC.2017.8288944.

[11] M.K. Hasan, M.M. Islam, M.M.A. Hashem, Mathematical model development to detect breast cancer using multigene genetic programming, in: 2016 5th Int. Conf. Informatics, Electron. Vis., IEEE, vol. 11, 2016, pp. 574–579, https://doi.org/10.1109/ICIEV.2016.7760068.
[12] M.M. Rahman, S. Nooruddin, K.M. Hasan, N.K. Dey, HOG+ CNN Net: Diagnosing COVID-19 and pneumonia by deep neural network from chest X-Ray images, SN Comput. Sci. 2 (2021) 1–15, https://doi.org/10.1007/s42979-021-00762-x.
[13] F. Varricchio, J. Iskander, F. Destefano, R. Ball, R. Pless, M.M. Braun, R.T. Chen, Understanding vaccine safety information from the vaccine adverse event reporting system, Pediatr. Infect. Dis. J. 23 (2004) 287–294, https://doi.org/10.1097/00006454-200404000-00002.
[14] K.M. Kahloot, P. Ekler, Algorithmic splitting: a method for dataset preparation, IEEE Access 9 (2021) 125229–125237, https://doi.org/10.1109/ACCESS.2021. 3110745.
[15] B.G. Marcot, A.M. Hanea, What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis? Comput. Stat. 36 (2021) 2009–2031, https://doi.org/10.1007/s00180-020-00999-9.
[16] Y. Bengio, Y. Grandvalet, No unbiased estimator of the variance of k-fold cross-validation, Adv. Neural Inf. Process. Syst. 16 (2003) 513–520. https://dl.acm.org/doi/.abs/10.5555/2981345.2981410.
[17] I. El Naqa, M.J. Murphy, What is machine learning? machine learning in radiation oncology 8 (2015) 3–11, https://doi.org/10.1007/978-3-319-18305-3_1.
[18] U.S. Shanthamallu, A. Spanias, C. Tepedelenlioglu, M. Stanley, A brief survey of machine learning methods and their sensor and IoT applications, in: 2017 8th Int. Conf. Information, Intell. Syst. Appl., IEEE, vol. 3, 2017, pp. 1–8, https://doi.org/10.1109/IISA.2017.8316459.
[19] S. Tangirala, Evaluating the impact of GINI index and information gain on classification using decision tree classifier algorithm, Int. J. Adv. Comput. Sci. Appl. 11 (2020) 612–619, https://doi.org/10.14569/IJACSA.2020.0110277.
[20] J. Han, J. Pei, H. Tong, Data Mining: Concepts and Techniques, third ed., Morgan kaufmann. 2012, ISBN 9780123814807.
[21] D. Liu, L. Clemente, C. Poirier, X. Ding, M. Chinazzi, J.T. Davis, A. Vespignani, M. Santillana, A Machine Learning Methodology for Real-Time Forecasting of the 2019-2020 COVID-19 Outbreak Using Internet Searches, News Alerts, and Estimates from Mechanistic Models, ArXiv Prepr (2020) 1–23, https://doi.org/10.48550/arXiv.2004.04019. ArXiv2004.04019.
[22] R. at al Sujath, J.M. Chatterjee, A.E. Hassanien, A machine learning forecasting model for COVID-19 pandemic in India, Stoch, Environ. Res. Risk Assess. 34 (2020) 959–972, https://doi.org/10.1007/s00477-020-01827-8.
[23] M.U. Rehman, A. Shafique, S. Khalid, M. Driss, S. Rubaiee, Future forecasting of COVID-19: a supervised learning approach, Sensors 21 (2021) 3322–3339, https://doi.org/10.3390/s21103322.
[24] S. Tiwari, S. Kumar, K. Guleria, Outbreak trends of coronavirus disease–2019 in India: a prediction, Disaster Med. Public Health Prep. 14 (2020) 33–38, https://doi.org/10.1017/dmp.2020.115.
[25] N.S. Punn, S.K. Sonbhadra, S. Agarwal, COVID-19 Epidemic Analysis Using Machine Learning and Deep Learning Algorithms, medRxiv (2020) 22–32, https://doi.org/10.1101/2020.04.08.20057679.
[26] M.A. Elaziz, K.M. Hosny, A. Salah, M.M. Darwish, S. Lu, A.T. Sahlol, New machine learning method for image-based diagnosis of COVID-19, PLoS One 15 (2020) 42–60, https://doi.org/10.1371/journal.pone. 0235187, e0235187.
[27] L.J. Muhammad, M. Islam, S.S. Usman, S.I. Ayon, Predictive data mining models for novel coronavirus (COVID-19) infected patients' recovery, SN Comput, Science 1 (2020) 1–7, https://doi.org/10.1007/s42979-020-00216-w.
[28] A. Fadli, A.W.W. Nugraha, M.S. Aliim, A. Taryana, Y.I. Kurniawan, W.H. Purnomo, Simple correlation between weather and COVID-19 pandemic using data mining algorithms, IOP Conf. Ser. Mater. Sci. Eng. 982 (2020) 12015–12029, https://doi.org/10.1088/1757-899X/982/1/012015. IOP Publishing.
[29] A.F. de Moraes Batista, J.L. Miraglia, T.H.R. Donato, A.D.P. Chiavegatto Filho, COVID-19 diagnosis prediction in emergency care patients: a machine learning approach, medRxiv 65 (2020) 8–16, https://doi.org/10.1101/2020.04.04.20052092.
[30] M.M. Hatmal, M.A.I. Al-Hatamleh, A.N. Olaimat, M. Hatmal, D.M. Alhaj-Qasem, T.M. Olaimat, R. Mohamud, Side effects and perceptions following COVID-19 vaccination in Jordan: a randomized, cross-sectional study implementing machine learning for predicting severity of side effects, Vaccines 9 (2021) 556–578, https://doi.org/10.3390/vaccines9060556.
[31] J. Avasarala, C.J. McLouth, L.C. Pettigrew, S. Mathias, S. Qaiser, P. Zachariah, VAERS reported new-onset seizures following use of Covid 19 vaccinations as compared to influenza vaccinations, Br. J. Clin. Pharmacol. 53 (2022) 1–5, https://doi.org/10.1111/bcp.15415.
[32] E. Saad, S. Sadiq, R. Jamil, F. Rustam, A. Mehmood, G.S. Choi, I. Ashraf, Novel extreme regression-voting classifier to predict death risk in vaccinated people using VAERS data, PLoS One 17 (2022) 1–29, https://doi.org/10.1371/journal.pone.0270327, e0270327.
[33] M.M. Ahamad, S. Aktar, M.J. Uddin, M. Rashed-Al-Mahfuz, A.K.M. Azad, S. Uddin, S.A. Alyami, I.H. Sarker, P. Liò, J.M.W. Quinn, Adverse effects of COVID-19 vaccination: machine learning and statistical approach to identify and classify incidences of morbidity and post-vaccination reactogenicity, medRxiv 22 (2021) 125–144, https://doi.org/10.1101/2021.04.16.21255618.
[34] R. Sujatha, B. Venkata Siva Krishna, J.M. Chatterjee, P.R. Naidu, N.Z. Jhanjhi, C. Charita, E.N. Mariya, M. Baz, Prediction of suitable candidates for covid-19 vaccination, Intell. Autom. Soft Comput. 9 (2022) 525–541, https://doi.org/10.32604/iasc.2022.021216.
[35] A. Alhazmi, E. Alamer, D. Daws, M. Hakami, M. Darraj, S. Abdelwahab, A. Maghfuri, A. Algaissi, Evaluation of side effects associated with COVID-19 vaccines in Saudi Arabia, Vaccines 9 (2021) 674–681, https://doi.org/10.3390/vaccines9060674.
[36] N.A. El-Shitany, S. Harakeh, S.M. Badr-Eldin, A.M. Bagher, B. Eid, H. Almukadi, B.S. Alghamdi, A.A. Alahmadi, N.A. Hassan, N. Sindi, Minor to moderate side effects of Pfizer-BioNTech COVID-19 vaccine among Saudi residents: a retrospective cross-sectional study, Int. J. Gen. Med. 14 (2021) 389–1401, https://doi.org/10.2147/IJGM.S310497.
[37] J. Chapin-Bardales, J. Gee, T. Myers, Reactogenicity following receipt of mRNA-based COVID-19 vaccines, JAMA 325 (2021) 2201–2202, https://doi.org/10.1001/jama.2021.5374.
[38] S. García, J. Luengo, F. Herrera, Data Preprocessing in Data Mining, Springer, Switzerland. 2015, pp. 59–139, https://doi.org/10.1007/978-3-319-10247-4.
[39] P. Sodhi, N. Awasthi, V. Sharma, Introduction to machine learning and its basic application in python, Proc. 10th Int. Conf. Digit. Strateg. Organ. Success (2019) 1354–1375, https://doi.org/10.2139/ssrn.3323796.
[40] D.M.W. Powers, Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation, ArXiv Prepr. ArXiv2010. 16061 (2020) 37–63, https://doi.org/10.48550/arXiv.2010.16061.