IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

CRITICAL DECISION SUPPORT SYSTEM AT HEALTHCARE USING MACHINE LEARNING

Tanusree Pathak, Susmit Sekhar Bhakta

Student, Student
Computer Science and Engineering(CSE)
Techno International Newtown, Kolkata, India

Abstract: Intensive Care Unit should be the most monitored and supervised area in a health-care sector. But, the unfortunate point is ICU is being monitored but not supervised always as patient-nurse ratio is not appropriate. Due to the lack of nurse's availability, insufficient number of beds in ICU, this problem happened. For most people, the ICU is a black box. According to the Society of Critical Care Medicine (SCCM), the five primary ICU admission diagnoses for adults are respiratory insufficiency/failure with ventilator support, acute myocardial infarction, intracranial hemorrhage or cerebral infarction, percutaneous cardiovascular procedures, and septicemia or severe sepsis without mechanical ventilation. Sometimes, it is very difficult to decide whether a patient needs to be transferred from a general ward to the ICU or not or even some medical persons may not decide whether the patient admitted into ICU needs to be transferred into the ICU or not. Misdiagnosis in the intensive care unit (ICU) is 50% more common than other areas. Not taking the right decision at the proper time may cause the patient's death. So, this is a very vital time to take action for making the right decision at the right time. In our paper, we have proposed a classification model by combining two most popular machine learning(ML) classification models i.e. logistic regression and Support vector machine(SVM). We have combined these two models by employing a most popular ensemble learning technique called Voting classifier. Our experiment proves that employing the ensemble technique of the Voting classifier outperforms all individual classification models. We have also shown a comparative study of logistic regression, SVM, Decision tree, Naïve-Bayes and their ensemble combinations. We have performed a performance test with the parameters of accuracy, sensitivity and specificity with our proposed model by using real-world dataset(from Kaggle). Feature selection also performed carefully to increase the performance and interpretability of our model and we achieved 99.65% of accuracy with our proposed model.

***Index Terms* - Critical Decision Support System(CDSS), Machine learning(ML), classification models, Logistic regression, SVM, ensemble learning, Decision tree, Naïve-Bayes, feature selection, accuracy, sensitivity, specificity, SCCM, Kaggle.**

I. INTRODUCTION

Critical decision Support Systems are a linker from patient's health data to a machine learning knowledge-based algorithm that helps physicians to make decisions in emergency [1]. These systems are made to help the medial persons by providing predictions, knowledge, necessary data whenever they need it [2]. CDSSs can be categorized into two parts which are knowledge-based and Artificial Intelligence model-based [4].

In this paper, we have worked on the part, transferring a patient to the ICU is a very difficult as well as confusing decision as it depends on so many factors. In this paper, the number of available beds in ICU, patient's heart rate, respiration level, number of available beds are some of the parameters from the dataset which we have considered to make the decision. It is very normal that a person's SpO2 level can suddenly increase or respiration level falls. In that situation, checking every parameter that has been mentioned above is very time consuming which can cause the death of that patient. Even when a patient is admitted to an ICU, there are so many changes of the parameters that are happening which are being controlled by some medical technologies. Sometimes, it can be overdosed or under-dosed which can also cause the death of the patient. At that point of time, the parameters need to be changed properly. So, here we are going to make a system which can support the total health-care system in the emergency by making the right decision by analyzing the parameters whether the patient needs the ICU or not and many more things using machine learning model so that it can be the decision support system in the critical moment and decrease the confusion level. Whenever a patient needs to be admitted to an ICU in a very critical situation, medical persons might get confused there and checking every parameter and making decisions after that takes a longer time and a confused decision can harm a patient. So, it is better to do the same by a system using a machine learning algorithm whether that patient needs to be admitted in ICU or not as CDSSs are used to reduce medical errors [3]. Similarly, an ICU patient might be released when all the body parameters are okay, and they can be shifted to the general ward. So, the system can check the same for both ICU and general ward patients. Then, ICU bed numbers as well as general ward bed numbers will be updated accordingly. All the patient's demographic details with the prediction by the model is stored in the database.

II. SOURCES OF DATA AND FEATURE SELECTION

For this study, we have taken the dataset of 25,493 patients from Kaggle that contains the Index and the vital signs of the patients. Vital signs include the features heart rate, respiration, SpO2, temperature.
To increase the performance of our model, we have dropped the column of "Index" which was not necessary to train the model. This dataset has been splitted into train and test data in the ratio of 0.2. The target column is "Priority" which indicates the patient's condition is "Normal" or "Abnormal". Getting admitted into an ICU or General ward in an emergency depends on the patient's condition("Priority").

III. THEORETICAL FRAMEWORK

In this paper we have used logistic regression, Decision Tree, SVM, Naïve-Bayes classifier and Ensemble method which are discussed below→

3.1. *Logistic Regression*– Logistic regression is a statistical model which is commonly used for binary classification problems. In this model, the response variable (also known as the dependent variable or target variable) takes on only two possible values. These values are usually labeled as 0 and 1. The main goal of logistic regression is to make an estimation of the probability that the response variable is equal to 1, given a set of predictor variables, These predictor variables are also known as independent variables or features. In this way, the logistic regression model makes predictions whether a particular observation belongs to one class or another.

The logistic regression model is based on the logistic function which is also known as the sigmoid function. This function maps any real-valued number to the range of [0, 1]. The logistic function can be defined as,

$$\sigma(m) = \frac{1}{1+\varepsilon^{-m}} \ldots\ldots\ldots\ldots\ldots\ldots(1)$$

In Equation1, m denotes the linear combination of the predictor variables. And its corresponding coefficients will be as,

$$m = \alpha_0 + \alpha_1 y_1 + \alpha_2 y_2 + \cdots + \alpha_n y_n \ldots\ldots\ldots\ldots(2)$$

In Equation2, α_0 is an intercept term. The predictor variables are $y_1, y_2, \cdots, y_n$ and their corresponding coefficients are $\alpha_1, \alpha_2, \cdots, \alpha_n$.
This logistic function transforms the linear combination of predictor variables into a probability value of [0, 1]. This model assumes that the probability of an event occurring i.e. the response variable taking on the value 1. It can be expressed as,

$$P(X = 1 | Y_1 = y_1, Y_2 = y_2, \cdots, Y_n = y_n) = \sigma(m) \ldots\ldots(3)$$

In Equation3, X is the response variable and $Y_1, Y_2, \cdots, Y_n$ are the predictor variables.
The logistic regression model is used to make prediction of the probability that the response variable is equal to 1, given a set of predictor variables. This is performed by plugging in the values of the predictor variables into the logistic function (sigmoid function) with the estimated coefficients which can be defined as,

$$\hat{P}(X = 1 | Y_1 = y_1, Y_2 = y_2, \cdots, Y_n = y_n) = y \ldots\ldots\ldots(4)$$

3.2. *Decision Tree*– A Decision tree is a tree-like structured model which represents decisions and their corresponding all possible consequences. This supervised machine learning algorithm can be used for both classification and regression purposes. Researchers opted for the Decision tree algorithm to solve complex problems as it is very easy to understand, interpret and visualize. This algorithm works by recursively splitting the dataset into subsets until a stopping criterion is fulfilled. The algorithm chooses the best splitting criterion to split the dataset at each of its decision nodes. The best splitting criterion can be defined as it maximizes the information gain(I_G)or minimizes the impurity of the subsets. Information gain can be expressed as following,

$$I_G(D,N) = H(D) - H(D|N) \ldots\ldots\ldots\ldots\ldots (5)$$

In Equation5, $I_G(D,N)$ is the information gain of splitting the dataset D on feature$N$, $H(D)$ is the entropy of the dataset D and $H(D|N)$ is the conditional entropy of the subsets created by splitting the dataset D on feature N.

Entropy can be calculated as,

$$H(D) = -\sum \quad p(j)\, log_2\, p(j) \ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

In Equation6, $p(j)$ is the proportion of the number of samples present in class j to the total number of samples. After the splitting of the dataset, the algorithm recursively applies the same process to each subset of the dataset until the stopping criterion is met. The stopping criterion is based on the depth of the tree, the number of samples present in the subset or the number of features in the subset. Once the decision tree is built, we can use it to make predictions of the output variable for a new input. For predicting the output variable, start at the root node of the tree and follow the path based on the input features. We will continue this process until we reach a leaf node which gives us the final prediction.

3.3 *Support Vector Machine(SVM)*– It is a supervised machine learning algorithm used for both classification and regression analysis. SVM is a non-probabilistic algorithm i.e. this model is able to directly learn a decision boundary which separates the different classes in the dataset.

In this model, the decision boundary is called the hyperplane. This algorithm tries to maximize the margin(the distance between the hyperplane and the closest data points from each class). The data points which are found to be closest to the hyperplane are called support vectors. These support vectors play a crucial role in defining the hyperplane.

For a binary classification problem, the SVM model tries to find the hyperplane that separates the two classes in a way such that the margin is maximized. The hyperplane can be defined as a linear function as below,

$$f(x) = w^T V_{in} + B \ldots\ldots\ldots\ldots (7)$$

In Equation7, x is the input vector, $w$is the weight vector, B is the bias term and $f(x)$ is the decision function that predicts the class label of x based on the sign of $f(x)$. If $f(x) > 0$ then x belongs to one class and if $f(x) < 0$ then x belongs to the other class.

So, the final formula SVM can be written as,

$$f(x) = sign(\sum_{j=1}^{n} \quad \alpha_j y_j K(x_j, x) + B) \ldots\ldots (8)$$

In Equation8, x is the input vector, y_j is the class label of the j^{th} training set, α_j are the Lagrange multipliers, $K(x_j, x)$ is the kernel function which measures the similarity between x_j and x in the feature space and B is the bias term.

The sign function returns the sign of the sum of the kernel evaluations plus the bias term, which determines the predicted class label of the input vector x.

3.4 *Naïve-Bayes Classifier*– Naïve Bayes classifier is very much famous among researchers for its simplicity and efficiency of this algorithm which is widely used for classification tasks. As its name depicts, This algorithm is based on the Bayes theorem(a probabilistic approach to make predictions). This classifier pre-assumes that all features are independent of each other which are given in the class label. For this assumption, this is called the "Naïve" Bayes classifier.

If a given set of features is $F = \{f_1, f_2, \cdots, f_n\}$ and label is L then, the Naïve Bayes classifier estimates the probability $P(L|F)$ is as,

$$P(L|F) = P(F|L) \times \frac{P(L)}{P(F)} \ldots\ldots\ldots\ldots\ldots (9)$$

In Equation9, $P(F|L)$is the conditional probability of observing the feature set Fgiven the class label L, $P(L)$ is the prior probability of the class label L and $P(F)$ is the probability of observing the feature set F.

The Naïve assumption that the features are independent of each other given the class label L which simplifies the conditional probability $P(F|L)$ as the product of individual feature probabilities are as,

$$P(F|L) = P(f_1|L) \times P(f_2|L) \times \cdots \times P(f_n|L) \ldots\ldots\ldots(10)$$

For classification of a new data point with feature set F, the classifier calculates the posterior probability for each class label L and assigns the label with the highest probability is as,

$$P(L = k_n|F) = \frac{P(F|L=k_n)\times P(L=k_n)}{\sum \quad P(F|L=k_m)\times P(L=k_m)} \ldots\ldots\ldots\ldots(11)$$

In Equation11, k_n is the ith class label and the sum is taken over all possible class labels k_m.

3.5. *Ensemble learning method* – Ensemble methods are very much known to the researchers as these methods make a visible improvement in the prediction and generalization by combining multiple models. Several methods are included in this ensemble like bagging, voting, stacking, boosting and many others.

In our paper we have used a voting classifier which is one of the most popular and simplest but useful ensemble methods. This voting classifier combines the predictions of the models(Decision tree, SVM etc.). These models must be trained independently on the same dataset before applying this voting classifier. Each of the models generates their own predictions and the final prediction is made by taking a vote from those predictions. Here, we have used “soft voting” in which the class with the highest average probability across all models is selected as the final prediction. The formula can be defined as follows,

$$ensemble_{voting} = argmax(sum(predicted_{class}[x] == y)\ for\ x\ in\ range(n_{models})) \ldots\ldots(12)$$

In Equation12, the final predicted class is $ensemble_{voting}$, predicted class of the xth model is $predicted_{class}[x]$, the class label is y and the total number of models in ensemble is n_{models}.

IV. Research Methodology

Data-driven, Knowledge-driven, Model-driven decision support systems are three categories in the modern healthcare area. In this framework, knowledge-driven and model-driven DSSs are combined. After an immense study has been done to gather knowledge in CDSS at healthcare, in this paper, we have launched a website that contains 3 webpages. The first or input web page takes input in .csv format that contains the patient's demographic information like - name, age etc. along with the vital signs (heart rate, respiration, SpO2, temperature) and currently available number of ICU as well as general ward beds. The second or output web page shows the prediction/result that the patient needs to get admitted in ICU or General ward or not which is a non-knowledge based CDSS. This model provides the prediction depending on a hybrid model of logistic regression and decision tree which is giving the accuracy of 99.65%. VotingClassifier is used here to fit both the classification algorithms. Also, this model provides a suggestion of an immediate action in an emergency that is a knowledge based CDSS depending on some predefined (IF-THEN-ELSE) rules. Critical Decision Support Systems can be divided into two categories which are Knowledge based and non-knowledge based CDSS. The third or database page contains the whole record of all the patients. It has the provision to delete a particular record of a patient as well. We have used python Flask to make the whole flow and SQLite as a database. In the background, a machine learning model has been designed which is a hybrid model of logistic regression and decision tree. We have taken the dataset of 25493 patients (vital signs) from Kaggle and then split, train and test our data. This system can be used by not only the medical persons but also any person in the world if they can measure their vital signs. Now-a-days, there are so many ways to find out heart rate, temperature, respiration, SpO2 level through pulse oximeter, thermometer and so many sensors which are easily available at market. Flow of our proposed model is well-designed through the model diagram.

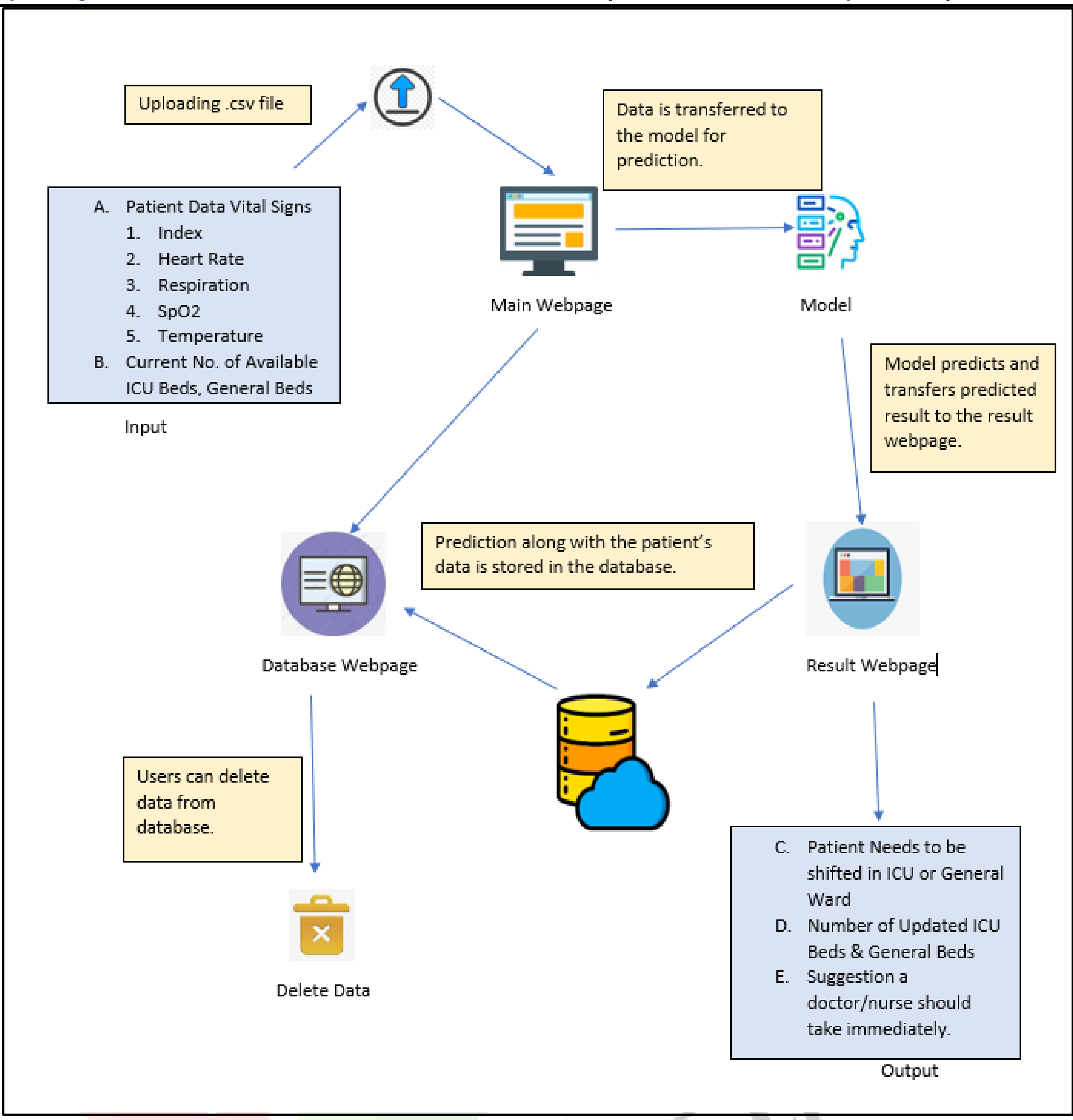

Figure 4.1: Model Diagram

V. RESULT AND DISCUSSION

As mentioned in the previous section, our model is an amalgamation of two classification algorithms which are logistic regression(LR) and decision tree(DT). The proposed model can be defined as follows,

$$proposed = ensemble_{voting}(LR, DT)............(13)$$

A deep experiment and comparison analysis has been done to get the outcome. Accuracy, specificity, sensitivity have been checked through confusion matrix. Total 25493 patient's dataset, containing vital signs has been taken. In this framework, a unique key has been assigned to recognise each and every patient and store their data or record. This experiment has been done on Support Vector Machine, Naïve-Bayes, Logistic Regression and Decision Tree classification algorithms. Different combinations on these algorithms have also been performed. A detailed result has been shown through Table1.

Models	Accuracy	Specificity	Sensitivity
Logistic Regression	97.39%	98.41%	93.99%
Decision Tree	98.64%	98.74%	98.32%
Support Vector Machine	97.68%	98.59%	94.67%
Naïve Bayes	97.49%	97.96%	95.94%
Support Vector machine + Naïve Bayes (ensembled)	97.57%	98.52%	94.42%
Logistic Regression + Decision Tree + Support Vector Machine + Naïve Bayes (ensembled)	98.51%	98.85%	97.38%
Logistic Regression + Decision Tree (ensembled) ← proposed model	99.65%	99.69%	99.49%

Table 5.1: A Comparative Study of Different Models

This comparison can be well understood through graphs in the form of bar chart as well.

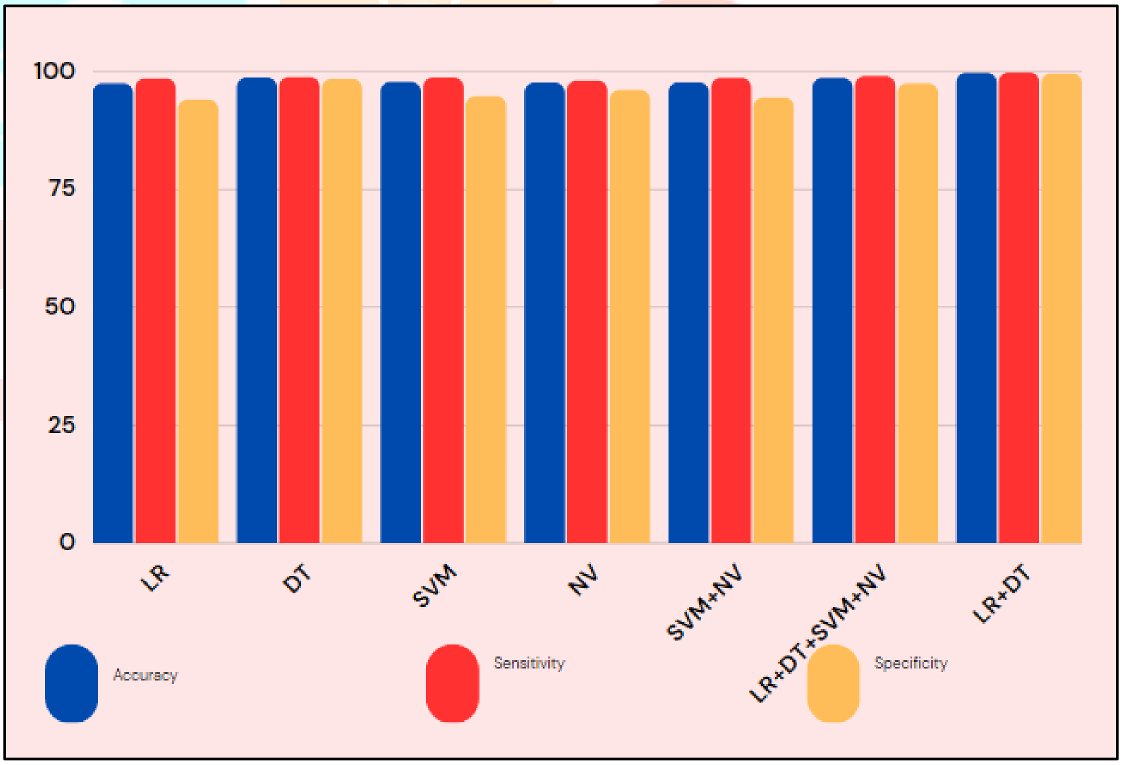

Figure 5.2: Comparison Analysis Chart

Healthcare is one of the most important areas in now-a-days. Decisions in this area must be as accurate as possible. In this paper, an immense study and experiment has been done to predict a patient's admission in ICU or General ward, predicted by the hybrid model of logistic regression and decision tree that provides the accuracy of 99%. Predictions by other models have been well explained in the above portions. Database management using SQLite, launching a user-friendly website to access all the data, updating the number of ICU and general ward beds are also covered through this paper. Website image is attached below.

Figure 5.3: Proposed CDSS Website

Further study for the extension of the model predictions in other areas and well upgradation of the website are the future scopes of this study. Overall, this is a well-designed machine learning model that predicts a quick and accurate action depending on a patient's vital signs.

VI. REFERENCES

[1] Beeler, Patrick Emanuel, David Westfall Bates, and Balthasar L. Hug. "Clinical decision support systems." Swiss medical weekly 144.5152 (2014): w14073-w14073.

[2] Moreira, Mário WL, et al. "A comprehensive review on smart decision support systems for health care." IEEE Systems Journal 13.3 (2019): 3536-3545.

[3] Khairat, Saif, et al. "Reasons for physicians not adopting clinical decision support systems: critical analysis." JMIR medical informatics 6.2 (2018): e8912.

[4] Sutton, Reed T., et al. "An overview of clinical decision support systems: benefits, risks, and strategies for success." NPJ digital medicine 3.1 (2020): 17.