

A System for Prediction and Analysis of the Appendicitis in Patients*

1st Sai Phanesh Reddy Ganta

dept. Information Science

University of North Texas

Denton, USA

2nd Sairam Yerramshetty

dept. Information Science

University of North Texas

Denton, USA

3rd Srilekha Nagu

dept. Information Science

University of North Texas

Denton, USA

4th Varun Reddy Kesari

dept. Information Science

University of North Texas

Denton, USA

saiphaneshreddyganta@my.unt.edu Sairamyerramshetty@my.unt.edu srilekhanagu@my.unt.edu Varunreddykesari@my.unt.edu

Abstract—The goal of this research is to create a system that can predict and analyze appendicitis in patients using Hyper-parameterized machine learning model and deep learning approaches i.e., CNN, LSTM and GRU. The obtained data will be statistically analyzed to uncover significant patterns and trends that might potentially be utilized to forecast the likelihood of appendicitis. The system also conducted the chi-square test and Spearman's correlation to check the relationship among the diseases. This technology will be used by medical practitioners to aid in the diagnosis and treatment of appendicitis patients, thereby improving patient life and state using data science and artificial intelligence.

Index Terms—appendicitis, artificial intelligence, data science, machine learning, deep learning, CNN, LSTM, GRU, Chi-square

I. INTRODUCTION AND STATEMENT OF THE PROBLEM

Appendicitis is an illness that can affect anyone at any time. Inflammation of the appendix, a tiny organ in the right lower abdominal quadrant, causes this illness. Serious complications, including appendix rupture and systemic infection, can result from ignoring an appendix.

In recent years, there has been a great of interest in developing prediction models for medical issues utilizing machine learning and deep learning methods. These models may be taught to look for certain patterns and trends in patient data that can indicate the presence or absence of a disease or condition. In order to better assess and predict appendicitis instances in patients, this study aims to apply machine learning and deep learning approaches. There may be important patterns and trends in the acquired data that can be utilized to predict the incidence of appendicitis. The use of this technology by doctors will improve the diagnosis and treatment of appendicitis, thereby benefiting the patients. Here we address the study's implications for the diagnosis and treatment of appendicitis, as well as its methodology, analysis, and outcomes. Understanding the potential triggers and causes of appendicitis is crucial before settling on a course of therapy. Predicting appendicitis occurrences and understanding their underlying causes is the focus of this work, which attempts to fill a knowledge vacuum in the existing literature. This system will built supervised machine learning techniques to

increase the accuracy of data classification and the reliability of previous studies. The system will be an essential resource for the healthcare sector, allowing for the delivery of personalized, cost-effective, and safe acute treatment. This endeavor will enhance patient outcomes by creating a reliable prediction model, and it may save healthcare expenses due to unnecessary procedures or misdiagnoses. The accuracy of machine learning models might be enhanced with better study into Python-based tools and approaches for data cleaning, preprocessing, and statistical analysis. The study aims to produce higher accuracy than the prior studies using different machine learning and data preprocessing techniques.

Whereas, this report's architecture generally consists of eight sections. The introduction and problem statement establish the tone for the report and show the seriousness of the subject under investigation. The literature review portion analyzes current literature on the issue thoroughly, revealing gaps that the study attempts to remedy. The study goals section describes the research aims and methods. The techniques utilized to acquire data are described in the data collecting section. The section on exploratory data analysis and hypotheses describes the findings of the study's exploratory data analysis and hypotheses. The statistical analysis done on the data is presented in the data analytics section. The data visualization and results section visually displays the study's findings. The conclusion and suggestions section highlights the study's significant results and makes recommendations for further research. Finally, in a consistent manner, the references section summarizes all of the sources referenced in the report.

A. Statement of Problem

The problem statement for the project is that there is an absence of comprehensive, data-driven research on appendicitis therapy, resulting in a lack of uniform standards or defined norms. The goal of this study is to evaluate and discover the underlying causes of appendicitis, as well as to see if it is possible to predict who will acquire the condition. Data-driven decision making will assist in the investigation of the disease's origins.

II. REVIEW OF LITERATURE

People between the ages of 10 and 30 have the highest incidence of developing acute appendicitis, making it a common occurrence at general surgery centers. Acute appendicitis occurs in around 7% of the population and requires surgery in 90% of cases. Clinical records were analyzed using machine learning algorithms and predictive data mining methods in an attempt to foresee the start of acute appendicitis in **Eason et al. (1955)**. Out of a total of 595 patients whose medical histories were examined, 348 (or 58.49%) were men and 247 (or 41.51%) were women. In study **Addiss et al. (1990)** Gradient boosted trees were shown to be the most effective prediction method, with an accuracy of 95.31 percent. Patients in need of immediate medical attention for appendicitis may benefit from this study's demonstration of machine learning's potential value in making such predictions.

The average age of the 7244 patients in study **Mijwil and Aggarwal (2022)** was 6.84 5.31 years, and 82.3% of the patients were male (5960/7244). The study **Mijwil and Aggarwal (2022)** found that the decision tree model was the most straightforward to understand of all the algorithms. This model successfully identified appendicitis in patients with a 93.97% area under the curve (AUC), 94.69% accuracy, 93.55% sensitivity, and 96.55% specificity. The AUC, accuracy, sensitivity, and specificity for detecting cases of uncomplicated appendicitis were 79.47%, 70.83%, 66.8%, and 81.88%, respectively. The study **Hayashi et al. (2021)**, trained an AI network with 6914 images from 50 videos. In the first test, three pediatric surgeons watched 10 videos and identified 50%, 20%, and 30% as "most," "partial," and "none," respectively. Three video groups had 35%, 35%, and 30% accuracy scores. The second test included six clips, and all participants, five pediatric residents and five pediatric intensive-care fellowship fellows, scored badly on detection and accuracy, calling all instances "none." Out of a total of 223,214 appendectomies, in study **Bunn et al. (2021)** researchers observed that 2,143 (0.96%) patients had sepsis after surgery. The area under the curve for each method (logistic regression, random forest decision trees, and extreme gradient boosting) was 0.70 (95% CI: 0.68-0.73), indicating that they all performed similarly. The area under the curve for support vector machines was 0.51 with a 95% confidence range of 0.50 to 0.52. Variable significance studies indicated preoperative congestive heart failure, transfusion, and acute renal failure as predictors of postoperative sepsis.

Appendicitis is a common surgical emergency that requires prompt diagnosis and treatment. Traditional diagnostic methods for appendicitis are based on physical examination, laboratory tests, and imaging studies. However, these methods have limited accuracy, and misdiagnosis of appendicitis can result in significant morbidity and mortality. Machine learning and deep learning techniques have been increasingly applied to predict the onset of appendicitis and improve diagnostic accuracy.

A study **Ting et al. (2010)** used machine learning algorithms

to predict acute appendicitis in emergency department patients. The authors analyzed 595 medical records and found that gradient enhanced trees had the highest accuracy of 95.3%. Another study **Akmese et al. (2020)**, used deep learning techniques to predict acute appendicitis on CT scans. The authors achieved a sensitivity of 98.5% and a specificity of 97.9% in detecting appendicitis on CT scans. In a retrospective study **Xia et al. (2022)**, the authors used machine learning algorithms to predict complicated appendicitis based on clinical and radiologic factors. The authors found that random forest models had the highest accuracy, with an area under the curve of 0.883. In another study **Park et al. (2020)**, the authors used a convolutional neural network to predict acute appendicitis based on ultrasound images. The authors achieved a sensitivity of 92.4% and a specificity of 93.8%.

III. OBJECTIVES OF THE STUDY

The objective of the study are listed below:

- To assess and derive conclusions from the existing research on the topic.
- To identify parameters from the raw data that may serve as early warning signs of appendicitis and evaluate the viability of using these indicators to predict the onset of appendicitis.
- To clean, preprocess and split the data for machine learning models and Neural Network models.
- To deploy state of the art machine learning and deep learning classifiers and test their accuracy using test data.
- To provide suggestions for the creation of universally accepted criteria or standards for the management of appendicitis will be made.

IV. DATA COLLECTION

The FDA and CDC developed Vaccine Adverse Event Reporting System (VAERS) to track and record information on adverse reactions to vaccines. Reports of incidents that may or may not be related to vaccinations are included in the VAERS collection, but distinguishing between them is challenging. The data include the different types of disease present or absent in the patient along with the vaccines applied to the patients. Following is the link to the dataset

- <https://www.kaggle.com/datasets/ayushggarg/covid19-vaccine-adverse-reactions>

V. EXPLORATORY DATA ANALYSIS (EDA) AND HYPOTHESES OF THE STUDY

To explore the data we conducted the statistical analysis of the data and to check the relationship of the features on each other, we graphically plot the charts from python famous library 'Matplotlib'. The data is downloaded in the form of csv files, that are converted into pandas data frame for performing various functions on the data which involves Exploratory Data Analysis and Data analytics. The figure 1, illustrate the first five rows of data frame that are used to represent the sample of the data frame. The raw data must have missing values and

errors that should be removed for analysis and machine learning modeling.

	VAERS_ID	RECVDAT	STATE	AGE_YRS	CAGE_YR	CAGE_MO	SEX	RPT_DATE	SYMPTOM_TEXT	DIED	...	Nausea	Pain	Pyrexia	Rash	T
0	916710	1/1/2021	MO	23.0	23.0	NaN	F	NaN	Acute appendicitis onset morning of 1/1/2021	NaN	...	False	False	False	False	
1	916720	1/1/2021	NY	23.0	23.0	NaN	M	NaN	Patient made statements that he was having a R.	NaN	...	False	False	False	False	
2	916741	1/1/2021	AR	68.0	68.0	NaN	F	NaN	on dec 22 I felt some myalgias, chills, fatigue.	NaN	...	False	True	False	False	
3	916772	1/1/2021	GA	55.0	55.0	NaN	M	NaN	Vaccine on 12/22/2020 and started feeling bad.	NaN	...	False	False	False	False	
4	916790	1/1/2021	TN	52.0	52.0	NaN	F	NaN	Flushing swelling increased heart rate proce.	NaN	...	True	False	False	False	

5 rows x 16 columns

Fig. 1. The graphical visualization of the first five rows of the dataframe

The data statistical analysis is carried out in figure 2, which represents the total count, mean, standard deviation, minimum, 25% value, 50% value, 75% value, and maximum value that are residing inside the column. For example, the AGE_YRS have maximum of 105 years, it concludes that patients age in the data set is maximum 105 years and minimum age of 1.08 years as shown in figure 2.

	VAERS_ID	AGE_YRS	CAGE_YR	CAGE_MO	HOSP_DAYS	NUM_DAYS	FORM_VERS
count	5.279000e+03	4343.000000	3154.000000	4.0	1151.000000	4519.000000	5279.000000
mean	9.694370e+05	57.122993	59.70038	0.0	3.286707	22.665634	1.996590
std	3.090224e+04	20.360390	20.79062	0.0	3.351753	680.312921	0.058299
min	9.167100e+05	1.080000	0.000000	0.0	1.000000	0.000000	1.000000
25%	9.412135e+05	40.000000	42.000000	0.0	1.000000	0.000000	2.000000
50%	9.733460e+05	56.000000	60.000000	0.0	2.000000	1.000000	2.000000
75%	9.950030e+05	74.000000	77.000000	0.0	4.000000	5.000000	2.000000
max	1.024853e+06	105.000000	105.000000	0.0	36.000000	36896.000000	2.000000

Fig. 2. Statistical analysis of the data frame

Whereas, the dataset contains 39 columns with Boolean values, 5 columns with decimal numbers, 2 columns with integer values, and 36 columns with string or categorical variables. The total memory usage of the dataset is approximately 1.9 MB on ram. This information of the data concluded that because of the 36 categorical columns, it is important to convert those columns into numerical to make the data set fit for the machine learning models and deep learning models.

Using the seaborn library, Figure 3 depicts a heatmap of the blank cells in a pandas dataframe df. The y-axis of the heatmap represents the columns of the dataframe, while the x-axis represents the rows. Yellow represents low values, while green and blue represent high ones in the "YlGnBu" colour map. Using the cbar kws parameter in the code, the "Missing Data" label is added to the colour bar.

Checking for missing values in a dataset is crucial because missing data can significantly alter the results of an analysis and the inferences that can be taken from the data. Mishandling missing values can cause inaccuracies or false conclusions to be drawn from statistical analyses. Furthermore, missing data can alter the shape of the distribution and the strength of associations between variables. The smaller the sample size, the less reliable the results will be, and the less likely it is that any important impacts will be overlooked. Therefore, before undertaking any statistical analysis or mod-

elling, it is necessary to properly identify and handle missing data to guarantee the validity and reliability of the results.

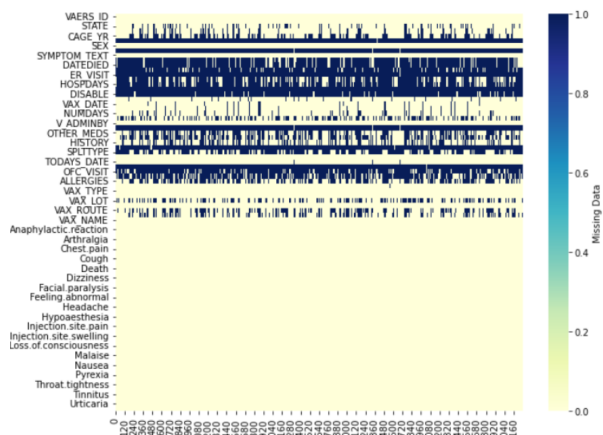


Fig. 3. Heat map representing the null values in the rows of each column

The data cleaning process will be explained in the data analytics part of the report. Checking for missing values in a dataset is crucial because missing data can significantly alter the results of an

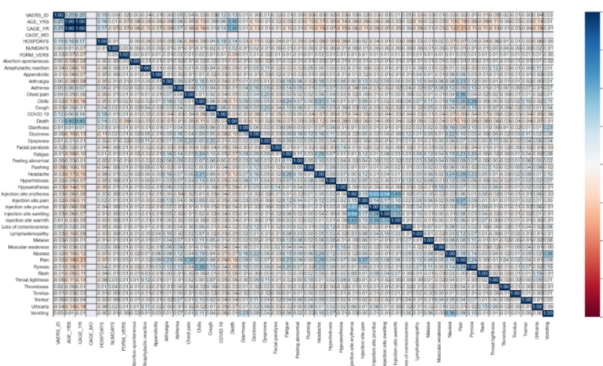


Fig. 4. Monotonic Correlation of the features

The strength and direction of a connection between two variables may be determined using Spearman's, a non-parametric statistical measure as shown in figure 4. It is used to assess whether or not two continuous variables are related monotonically, with one increasing or decreasing proportionally to the other. There is no monotonic relationship when the value is 0, and there is a perfect negative monotonic relationship when the value is -1. Spearman's test is a statistical correlation coefficient used in exploratory data analysis to identify monotonic relationships between variables, even when the connection is non-linear or the variables are not normally distributed.

A. Hypotheses

One other method is chi-square test for checking the null hypotheses and alternate hypotheses of relationship between two columns or features that have binary instances.

- The null hypothesis states that there is no effect of features on each other.

- The alternative hypothesis suggests that features have a significant effect on each other.

The p-value obtained from the analysis is greater than the commonly used significance level of 0.05. This suggests that there is insufficient evidence to reject the null hypothesis and accept alternative hypotheses to conclude that the two features are independent. The system uses the following algorithm for conducting the hypotheses test:

- If the p-value is less than or equal to 0.05, the null hypothesis is rejected, and it is concluded that there is a significant relationship between the column and Appendicitis, and the column name is added to the significant columns list.
- If the p-value is greater than 0.05, the alternative hypothesis is rejected, and it is concluded that there is no significant relationship between the column and Appendicitis, and the column name is added to the norelation columns list.

Following is the list of 19 features which accept the alternative hypotheses and have effect on target: ['RECVDATE', 'STATE', 'CAGE_YR', 'HOSPDAYS', 'RECOVD', 'NUMDAYS', 'LAB_DATA', 'V_ADMINBY', 'TODAYS DATE', 'symptom list', 'VAX LOT', 'VAX DOSE SERIES', 'Appendicitis', 'Death', 'Dyspnoea', 'Lymphadenopathy', 'Nausea', 'Pain', 'Vomiting']. The figure 5 plot shows the probability density function for Monotonic correlated column, allowing us to visually compare their distributions and relation.

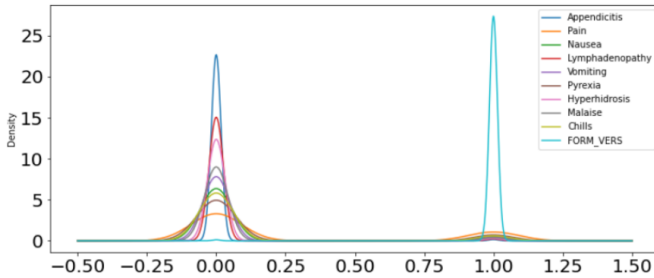


Fig. 5. Distribution of Monotonic Correlated features with the target column i.e., Appendicitis

VI. DATA ANALYTICS

For the data analytic part of the system, we will split our data set into train and test dataset with the 80:20 ratio respectively. The 80% data will be used to train the hyperparameterized models and then tested using the 20% of the unseen data. The system split the dataset into training and testing sets using the train test split function from the sklearn.model selection module. In the first step, the system separates the target variable 'Appendicitis' from the feature matrix X by dropping the 'Appendicitis' column from X and assigning column 'Appendicitis' to y. Then, the code calls train test split function, passing X and y as the feature and target matrices, respectively. The test size parameter is set to 0.2, indicating that the testing dataset should be 20% of the total dataset. The random state parameter is set to 42, which ensures

that the data is split in a consistent way every time the code is run, making the dataset free of biasing.

Followings are the shapes of the split datasets i.e., training and testing datasets:

- Shape of Xtrain: (1869, 9)
- Shape of y train: (1869,)
- Shape of X test: (468, 9)
- Shape of y test: (468,)

A. Hyper-Parameterized Machine Learning Modeling

Finding the optimal set of hyperparameters that maximises the model's accuracy is called "parameter tuning," and it's a crucial part of developing machine learning models. All five models' hyperparameters have been optimised in the provided code for optimal performance on the test data.

Following is the list of machine learning models and Hyper parameters to be implemented in this study as shown in figure 6:

- Logistic Regression: The value of 0.001 for the hyperparameter 'C' was determined to be optimal for Logistic Regression. The model was trained using this value, yielding an accuracy of 0.9930452609998422 during training and 0.9935897435897436 during testing.
- SVM: The optimal values of 'C' = 0.001 and 'kernel' = 'linear' were discovered for SVM's hyperparameters. Using these hyperparameters, the model was able to replicate the Logistic Regression model's training and testing accuracy.
- Random Forest: The optimal values of 'max depth' and 'n estimators' for Random Forest were determined to be 0 and 10, respectively. Using these hyperparameters, the model was able to replicate the training and testing accuracy of the prior two models.
- Decision Tree: The optimal values of 'max depth' for the Decision Tree algorithm were discovered to be 'None' and 'min samples split' was determined to be '2'. These hyperparameters were used to train the model, and it replicated the accuracy of the prior three models in both the training and testing phases.
- Naive Bayes: The optimal value of the Naive Bayes hyperparameter 'var smoothing' was determined to be 1e-09. The model's training and test accuracies were significantly lower than those of the other models, coming in at 0.1546271738039598 and 0.17735042735042736, respectively.

B. Deep Learning Modeling

The system used CNN (Convolution Neural Networks), LSTM (Long Short-Term Memory Networks) and GRU (Gated Recurrent unit) model for prediction of the appendicitis in the patients based on the data set that was fed to the machine learning models but it was concluded that CNN model needs to convert the data into 1-Dimensional Array, thus the data is preprocessed again for CNN model. The training and testing of the models is done on the training and testing data.

```

Logistic Regression
Best hyperparameters: {'C': 0.001}
Training accuracy: 0.9930452609998422
Test accuracy: 0.9935897435897436

SVM
Best hyperparameters: {'C': 0.001, 'kernel': 'linear'}
Training accuracy: 0.9930452609998422
Test accuracy: 0.9935897435897436

Random Forest
Best hyperparameters: {'max_depth': None, 'n_estimators': 10}
Training accuracy: 0.9930452609998422
Test accuracy: 0.9935897435897436

Decision Tree
Best hyperparameters: {'max_depth': None, 'min_samples_split': 2}
Training accuracy: 0.9930452609998422
Test accuracy: 0.9935897435897436

Naive Bayes
Best hyperparameters: {'var_smoothing': 1e-09}
Training accuracy: 0.1546271738039598
Test accuracy: 0.17735042735042736

```

Fig. 6. Hyper Parameterized Modeling

- CNN: As shown in figure 7, CNN model consists of multiple layers, each with its own distinct properties. The first layer is a Conv1D layer with 64 filters and a kernel size of 3,

Model: "sequential_3"

Layer (type)	Output Shape	Param #
conv1d_2 (Conv1D)	(None, 79, 64)	256
max_pooling1d_2 (MaxPooling1D)	(None, 39, 64)	0
conv1d_3 (Conv1D)	(None, 37, 32)	6176
max_pooling1d_3 (MaxPooling1D)	(None, 18, 32)	0
flatten_1 (Flatten)	(None, 576)	0
dense_4 (Dense)	(None, 128)	73856
dropout_4 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 1)	129

Total params: 80,417
 Trainable params: 80,417
 Non-trainable params: 0

Fig. 7. CNN Model Summary

which accepts an input tensor of shape (None, 81, 1) and returns a tensor of shape (None, 79, 64). This layer's parameters number 256. This layer's output is then sent to a MaxPooling1D layer with a pool size of 2, which produces a tensor with the shape (None, 39, 64). The model then contains a Conv1D layer with 32 filters and a kernel size of 3, which generates a (None, 37, 32)-shaped tensor. This layer's parameters total 6,176. The output of this layer is then transmitted to a MaxPooling1D layer with a pool size of 2, which produces a (None, 18, 32) tensor. The model then includes a Flatten layer, which transforms the output of the previous layer into a tensor with the shape (None, 576). The following layer is a Dense layer with 128 neurons that accepts the flattened tensor as input and outputs a tensor with the shape (None, 128). This layer's parameters total 73,856. The model also contains a Dropout layer, which arbitrarily sets a fraction of the input units to 0 at each training update. This layer's parameters are not trainable. The model concludes with

a Dense layer containing 1 neuron, which accepts the output of the previous layer as input and outputs a (None, 1) tensor. This layer's parameters total 129. The model has a total of 80,417 trainable parameters.

- LSTM: The model consists of four sequential layers and six layers in total as shown in figure 8. The first layer is a 64-unit LSTM layer that receives an input tensor with the shape (None, 81, 1) and produces an output tensor with the shape (None, 81, 64). There are 16,896 trainable parameters for this layer.

Model: "sequential_4"

Layer (type)	Output Shape	Param #
lstm_3 (LSTM)	(None, 81, 64)	16896
dropout_5 (Dropout)	(None, 81, 64)	0
lstm_4 (LSTM)	(None, 81, 64)	33024
dropout_6 (Dropout)	(None, 81, 64)	0
lstm_5 (LSTM)	(None, 64)	33024
dropout_7 (Dropout)	(None, 64)	0
dense_6 (Dense)	(None, 1)	65

Total params: 83,009
 Trainable params: 83,009
 Non-trainable params: 0

Fig. 8. LSTM Model Summary

The second layer is a dropout layer that sets a random fraction of input units to 0 at each training update. No trainable parameters exist. The third layer is a 64-unit LSTM layer that outputs a tensor with the shape (None, 81, 64) based on the output of the second layer. There are 33,024 trainable parameters for this layer. Additionally, the fourth layer is a dropout layer with no trainable parameters. The fifth layer is a final 64-unit LSTM layer that receives the output of the previous layer and emits a (None, 64) tensor. There are 33,024 trainable parameters for this layer. The sixth and final layer is a dense layer with one neuron that accepts the output of the previous layer as input and outputs a (None, 1) tensor. There are 65 trainable parameters in this layer.

- GRU: The GRU model as shown in figure 9, consists of only two layers i.e., a GRU layer with 64 units that accepts an input of shape (None, n steps, n features) and outputs a tensor of shape (64). This layer's parameters number 12,864. The second layer is a Dense layer with a single output neuron that receives the output of the previous layer and generates a (None, 1) tensor. This stratum has sixty-five parameters. The model contains 12,929 trainable parameters in total.

Model: "sequential_5"		
Layer (type)	Output Shape	Param #
=====		
gru_1 (GRU)	(None, 64)	12864
dense_7 (Dense)	(None, 1)	65
=====		
Total params: 12,929		
Trainable params: 12,929		
Non-trainable params: 0		

Fig. 9. GRU Model Summary

VII. DATA VISUALIZATION AND RESULTS

It is concluded that the parameter tuning was successful in this project. According to Figure 10, the Decision Tree, Random Forest, Support Vector Machine (SVM), and Logistic Regression models achieved an accuracy of 99.36%, which is a very high accuracy rate than the prior studies. On the other hand, the Naive Bayes model did not perform well in this case and only achieved an accuracy of 17.74%. These results suggest that the selected models are capable of accurately predicting the outcomes of the project, and that the tuning of the model's parameters had a significant positive impact on their performance.

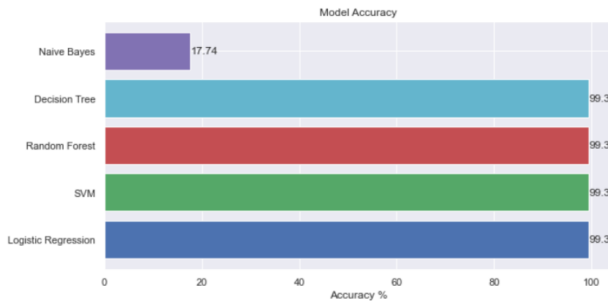


Fig. 10. Results of Machine Learning Models

The training pattern of a CNN model with 10 epochs is shown in Figure 11. During training, the model achieved a high degree of precision, with values ranging from 0.9806 to 0.9924. Training loss decreased from 2.1672 in the first epoch to 0.0467 in the last epoch, indicating that the model improved with each epoch. Similarly, the validation loss decreased from 0.5987 in the first epoch to 0.0752 in the last. This demonstrates that the model is applicable to novel, unobserved data. Notably, validation accuracy was consistently high, indicating that the model did not suffer from data overfitting. This model appears to have adequate efficacy overall.

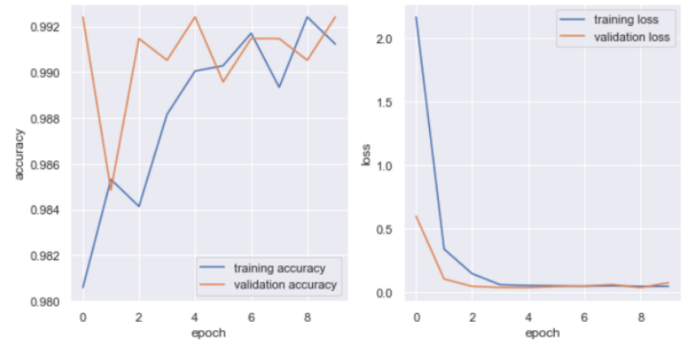


Fig. 11. History of CNN for 10 Epochs

Figure 12 demonstrates that after the first epoch, the model LSTM obtained an accuracy of 98.74% on the training data and 99.24% on the validation data. For the remaining epochs, the model's accuracy on the training data remained constant at 99.17%, while the validation loss increased from 0.0446 to 0.0459 before decreasing to 0.0456 in the final epoch. Throughout the training procedure, the validation accuracy remained stable at 99.24%. Figure 13 represents the GRU model's training and validation history. On a dataset, it has been trained for 10 epochs with 132 training steps per epoch. Throughout the first epoch, the model achieved a training loss of 0.1395 and an accuracy of 0.9915 on the training set, whereas the validation loss was 0.0478 and the validation accuracy was 0.9924 on the validation set.

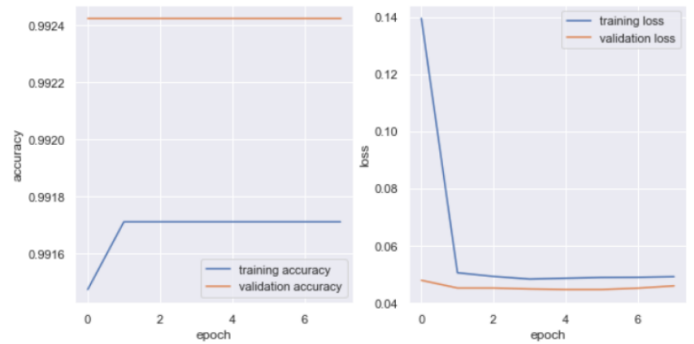


Fig. 12. History of LSTM for 10 Epochs

In subsequent epochs, the performance of the model improved, as indicated by a decrease in loss values. Consistency in the model's accuracy across epochs suggests that it does not overfit the data. In epoch 4, validation loss is at its lowest, 0.0448, and validation accuracy is at its maximum, 0.9924. However, performance across the remaining epochs is relatively stable, indicating that the model is not sensitive to the number of training epochs.

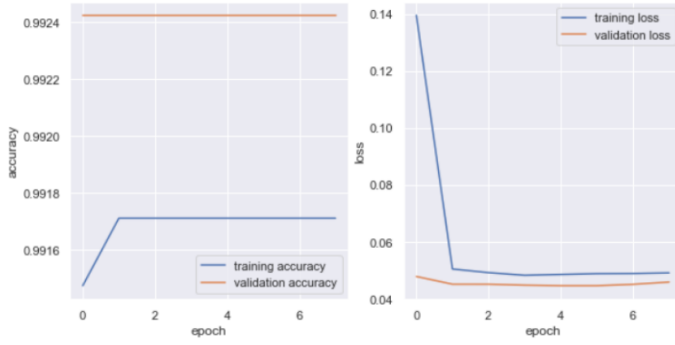


Fig. 13. History of GRU for 10 Epochs

The figure 14, represents that the CNN, LSTM and GRU models achieved accuracy of 99.24% for all of the models when tested on unseen data to predict the appendicitis.

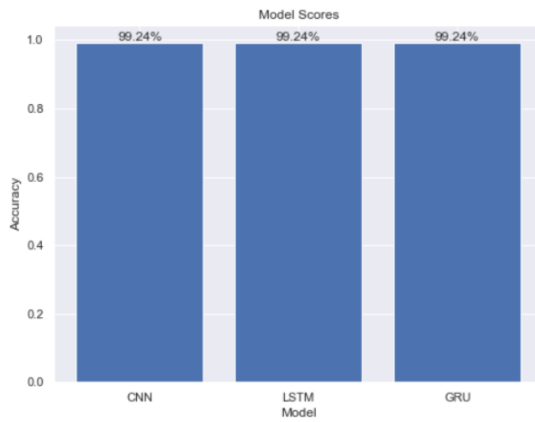


Fig. 14. Accuracies of Deep Learning Models

VIII. CONCLUSION

In the absence of exhaustive, data-driven research on appendicitis therapy, there are no uniform standards or established norms. The purpose of this investigation was to evaluate and identify the underlying causes of appendicitis, as well as to determine whether it is possible to predict who will develop the condition. The study's objectives included evaluating and deriving conclusions from existing research on the subject, identifying parameters from raw data that may serve as early warning signs of appendicitis, evaluating the viability of using these indicators to predict the onset of appendicitis, deploying state-of-the-art machine learning and deep learning classifiers, and providing suggestions for the creation of universally accepted criteria or standards for the management of appendicitis. The dataset utilised for the investigation was the FDA and CDC-developed Vaccine Adverse Event Reporting System (VAERS). The exploratory data analysis revealed that the data frame contains absent values and defects, necessitating its cleaning prior to further examination. The cleansing procedure will be described in the section of the report devoted to data analytics. Using monotonic correlation and Spearman's test and Chi-square, the intensity and direction of the relationship between variables were determined. In the future, the results

of this study may contribute to the development of universally accepted criteria or standards for the treatment of appendicitis. We developed a system that uses both traditional machine learning algorithms and deep learning models to predict appendicitis in patients. The dataset was divided into training and testing sets, and the hyperparameters of five machine learning models were optimised: Logistic Regression, SVM, Random Forest, Decision Tree, and Naive Bayes. The Logistic Regression, SVM, and Random Forest models performed wonderfully, whereas the Naive Bayes model was inaccurate. Additionally, we implemented a CNN model, which required the data to be preprocessed into a one-dimensional array. The CNN model consisted of multiple layers, including Conv1D and MaxPooling1D layers, and attained high precision. Overall, our system provides a dependable method for predicting appendicitis in patients by employing both conventional and deep learning techniques.

REFERENCES

- Addiss, D. G., Shaffer, N., Fowler, B. S., and Tauxe, R. V. (1990). The epidemiology of appendicitis and appendectomy in the united states. *American journal of epidemiology*, 132(5):910–925.
- Akmese, O. F., Dogan, G., Kor, H., Erbay, H., and Demir, E. (2020). The use of machine learning approaches for the diagnosis of acute appendicitis. *Emergency Medicine International*, 2020.
- Bunn, C., Kulshrestha, S., Boyda, J., Balasubramanian, N., Birch, S., Karabayir, I., Baker, M., Luchette, F., Modave, F., and Akbilgic, O. (2021). Application of machine learning to the prediction of postoperative sepsis after appendectomy. *Surgery*, 169(3):671–677.
- Eason, G., Noble, B., and Sneddon, I. N. (1955). On certain integrals of lipschitz-hankel type involving products of bessel functions. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 247(935):529–551.
- Hayashi, K., Ishimaru, T., Lee, J., Hirai, S., Ooke, T., Hosokawa, T., Omata, K., Sanmoto, Y., Kakiyara, T., and Kawashima, H. (2021). Identification of appendicitis using ultrasound with the aid of machine learning. *Journal of Laparoendoscopic & Advanced Surgical Techniques*, 31(12):1412–1419.
- Mijwil, M. M. and Aggarwal, K. (2022). A diagnostic testing for people with appendicitis using machine learning techniques. *Multimedia Tools and Applications*, 81(5):7011–7023.
- Park, J. J., Kim, K. A., Nam, Y., Choi, M. H., Choi, S. Y., and Rhie, J. (2020). Convolutional-neural-network-based diagnosis of appendicitis via ct scans in patients with acute abdominal pain presenting in the emergency department. *Scientific reports*, 10(1):1–9.
- Ting, H.-W., Wu, J.-T., Chan, C.-L., Lin, S.-L., and Chen, M.-H. (2010). Decision model for acute appendicitis treatment with decision tree technology—a modification of the

alvarado scoring system. *Journal of the chinese medical association*, 73(8):401–406.

Xia, J., Wang, Z., Yang, D., Li, R., Liang, G., Chen, H., Heidari, A. A., Turabieh, H., Mafarja, M., and Pan, Z. (2022). Performance optimization of support vector machine with oppositional grasshopper optimization for acute appendicitis diagnosis. *Computers in Biology and Medicine*, 143:105206.