ISSN : 2582-7464

Computational Intelligence and Machine Learning

Volume No. 5

Issue No. 2

May - August 2024

ENRICHED PUBLICATIONS PVT.LTD

JE - 18,Gupta Colony, Khirki Extn,
Malviya Nagar, New Delhi - 110017.
E- Mail: info@enrichedpublication.com
Phone :- +91-8877340707

ISSN : 2582-7464

Computational Intelligence and Machine Learning

Aims and Scope

The primary objective of the Computational Intelligence and Machine Learning is to serve as a comprehensive, open-access platform that is dedicated solely to facilitating the progress and advancement of the field of Artificial Intelligence & Machine Learning by -

offering gifted and talented researchers engaged within the domain of Artificial Intelligence & Machine Learning a unique setting for them to get their work published and elevate their reputations/standing within the global community, as well as,

providing professionals, students, academics, and scholars free access to the latest and most advanced research outcomes, findings, and studies, being carried out in the field of Artificial Intelligence & Machine Learning, all across the world.

Related Topics

Soft computing

Fuzzy Logic

Artificial Neural Networks

Evolutionary Computing

Artificial Intelligence and Machine Learning

Artificial Immune Systems

Probabilistic Methods

Cognitive Robotics

Data mining

Computational Intelligence Methods for Bioinformatics and Biostatistics

ISSN : 2582-7464

Other emerging topics in Computational Intelligence

Nanobioscience

Information Forensics and technology

Nanotechnology

Cybersecurity

Big Data

Bioengineering and Biotechnology

Computational Neuroscience

ISSN : 2582-7464

Advisor

Editor-in-chief

Editorial Board Members

ISSN : 2582-7464

ISSN : 2582-7464

ISSN : 2582-7464

Computational Intelligence and Machine Learning

(Volume No. 5, Issue No. 2, May - August 2024)

Contents

Decision Tree Algorithm for Predicting Student Performance Based on Psychological Tests

San A. Limbong1, Estomihi R. Sirait2, Cristina S. Hasibuan3, Mario E. S. Simaremare4*
1,2,3,4 Institut Teknologi Del, Indonesia

A B S T R A C T

It is essential to consider the psychological aspect of selecting new students to determine the success of prospective students. In this paper, we propose an approach to predict student performance based on their psychological test scores using the Decision Tree algorithm. The dataset used in this study was taken from the student admission process at the Institut Teknologi Del.
The admission dataset contains the scores of psychological tests and the Grade Point Average (GPA) of classes 2019, 2020, and 2021. Each class has its own attribute set. Therefore, we came up with two approaches. The first approach was to use as many records as possible, and the opposite of the second was to utilize more features.
Our results showed that the first approach was slightly better. The MAE value was 0.3654 to 0.4568. Moreover, none of the psychological test attributes strongly correlate to GPA and hence do not guarantee student performance.

Keywords : Decision Tree, Machine Learning, Psychological Test

INTRODUCTION

Student learning success is inseparable from the influence of various factors, such as the learning environment or other factors (both internal and external). This is supported by [1], which concluded that one factor that significantly influences student success is motivation and whether the student has learning talent. Another illustration is that student motivation and satisfaction positively correlate with student learning outcomes.

A psychological test is a test used to measure individual differences and individual reactions on different occasions. Psychological tests are used to get candidates according to the abilities expected to achieve organizational needs [2]. The application of psychological tests is very important to determine the suitability or eligibility of the individual for the organization or institution.

The Institut Teknologi Del (IT Del) student admission process includes academic tests after which a psychological test is used to measure the level of ability of prospective students in the social, emotional, personality, and potential fields. Psychological tests are provided at each entrance after the academic tests are carried out. The psychological test measurement is intended to see whether the candidate can adapt to the campus lifestyle.

Based on this, the admission is decided by examining both the psychological and academic aspects. The psychological test conducted at IT Del has several measurement categories such as General Intelligence Test (TIU), Emotional Stability, Work Achievement, Work Tempo, Accuracy, Consistency, Endurance and Intellectual Quotient (IQ) or Work Attitude and Intelligence. Furthermore, each aspect of the psychological test will be measured according to the applicable rating scale. Psychological test aspects are calculated using a letter scale (grade) with 2 format, the first format includes Very Poor (KS), Poor

(K), Somewhat Poor (AK), Fair/Average, Somewhat Good (AB), Good (B) and Excellent (BS) while the second format includes Poor (K), Moderately Poor (S-), Moderate (S), Sufficiently Poor (C-), Sufficient/Adequate (C), Sufficiently Good (C+), Moderately Good (B) and Good (B).

The candidate's eligibility is decided by the Head of the Study Program where the candidate applies. In the Information Systems study program, the requirements for prospective students who are considered eligible are measured through the General Intelligence Test (TIU) with a range of scores greater than or equal to 10 (TIU >= 10), IQ greater than or equal to 105 (IQ>= 105). For the results of each aspect that has a Somewhat Poor (AK) value, there can be no more than 3, and there are no aspects with a Poor (K).

This assessment has the potential to result in human error, as well as subjective decisions. In other words, the assessment can override the application of the prerequisite scale that has been determined by considering other aspects. Based on the problems above, we need a machine learning model that can provide predictions for prospective students based on their academic achievements when they enter IT Del. The model will work by comparing aspects of the psychological test assessment and comparing the Grade Point Average (GPA) of the previous students while participating in active lectures at IT Del. The GPA has been studied through previous research [2], which predicted student GPA based on first-semester results. This study used computer science course data, followed by grades from six courses, one laboratory result, and GPA in the graduation year. The method used in this study is the Generalized Linear Model, Deep Learning, and Decision Tree.

This research will focus on developing a machine learning model using a decision tree algorithm to leverage the flexibility and interpretability of the algorithm while benefiting from the improved accuracy and generalization capabilities of machine learning. The machine learning algorithms help overcome the limitations of decision trees by optimizing the tree structure, reducing overfitting, and capturing complex patterns in the data. Thus, using machine learning models using decision tree will help predict the right prospective new Del Technology Institute students according to their academic achievements.

LITERATURE REVIEW

Psychology test

Psychological tests have various data collection techniques, such as tests, interviews, case studies, behavioral observations, and other procedures [2]. Based on research [3], the implementation of psychological data collection tests that are commonly used, such as paper and pencil tests, objective and essay tests, standard and non-standard tests, individual and group tests, verbal or nonverbal tests, personality tests, interest tests, aptitude tests, achievement test, intelligence test, and vocational test.

In applying psychological tests, several characteristics must be met [4], namely validity and reliability. Validity is the degree to which the measurement is accurate through a psychological test. At the same time, reliability is the level of consistency with the tests performed. This aims to set the level of Stability and relativity of the tests performed.

Reliability is dependability, Stability, consistency, predictability, and accuracy. If it meets the reliability criteria, then the assessment results from the test can be interpreted as reliable.

Decision Tree

Supervise Learning algorithm, which can be used for regression or classification. In other words, the decision tree can be used for numerical and categorical data. The decision tree algorithm works like a tree, where class labels are leaves and features (or conditions) are branches. Decision trees are used to deal effectively with large non-linear data sets. The decision tree observes the characteristics of an object and trains the model in a tree structure to predict future data to produce meaningful continuous outputs.

Continuous output means that the output/result is not discrete. It is not represented simply by a discrete set of known numbers or values. The decision tree divides the dataset into smaller subsets, and decisions are formed in stages [5]. Decision trees are used to deal effectively with large non-linear data sets. Besides that, decision tree algorithms are easy to understand, interpret and visualize [6]. Evaluation Metrics Based on research [7], the evaluation metrics used to measure forecasting errors and assess predictive models in the regression model are MAE, MSE, RMSE, and MAPE. In what follows, we'll elaborate more on research-based error evaluation metrics used in this research.

1) MSE (Mean Squared Error)

MSE is the difference - the average square of the difference between the predicted and actual values [8]. The greater the MSE value, the worse the model performance will be, and vice versa.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (Xi^2 - X'i^2)$$

2) MAE (Mean Absolute Error)

MAE measures how well the regression model predicts the actual target value. MAE is calculated by taking the average absolute error (model prediction minus the true value). The greater the MAE value, the worse the model's performance in predicting the target value, and vice versa. MAE gives less weight to outliers [7].

$$MAE = 1/n\sum_{i=1}^{n} |Xi - X'i|$$

3) MAPE (Mean Absolute Percentage Error)

MAPE measures the average absolute percentage error between predicted and true values. The lower the MAPE value, the smaller the prediction error in the model.

$$MAPE = 1/n\sum_{i=1}^{n} |Xi - X'i/Xi| * 100$$

Further description of the evaluation in research will display the evaluation results. However, the evaluation using MAE is more emphasized. This is as described in research [7], suggesting that the MAE algorithms are more appropriate for determining the accuracy of predictions. In line with that study, research [9] suggests that the MAE evaluation metrics are better used to compare performance between different regression models.

Correlations Coefficient analysis

Correlation coefficient analysis is a statistical technique used to measure the strength and direction of the relationship between two variables. It provides a numerical value that indicates the extent to which the variables are linearly related. The following is interpretation of the range of correlations coefficient analysis.

Table 1: Correlations Coefficient analysis [10]

Positive	Negative	Interpretation
+1.00	-1.00	Perfect
+0.80 to +0.99	-0.80 to -0.99	Very Strong
+0.60 to +0.79	-0.60 to -0.79	Strong
+0.40 to +0.59	-0.40 to -0.59	Moderate
+0.20 to +0.39	-0.20 to -0.39	Weak
+0.01 to +0.19	-0.01 to -0.19	Very Weak

Positive correlation refers to the relationship between two variables where a change in the value of one variable is followed by a change in the same direction in the other variable. Conversely, negative correlation occurs when a change in the value of one variable is followed by a change in the opposite direction in the other variable.

METHODOLOGY

Data

The dataset will be taken from the student admission process at IT Del, which is presented with data on psychological test results and the GPA of active students majoring in Informatics and Information Systems until 2022/2023. In particular, the data used are for the 2019, 2020, and 2021 batches. The following is a description of the data used and the features/variables in it.

From the results of the Analysis carried out above, there are three categories related to the availability of column attributes in the dataset: Available, Limited, and None. Restricted categories will be removed in the future due to the limited availability of records in the 2020 dataset, which only contains 14 of the 50 records in the 2020 dataset. Also, another reason why these records will be deleted is the format is different from the rest of the 2020 data set. Thus it is not possible to do a combination of column attributes.

Table 2: Comparison of attribute datasets

Attribute	2019	2020	2021
Seq	Available	Available	Available
GPA1	Available	Available	Available
GPA2	Available	Available	Available
GPA3	Available	Available	Available
GPA4	Available	Available	None
GPA5	Available	None	Available
GPA6	Available	None	Available
GPA	None	None	Available
Batches	None	None	Available

TIU	None	Available	Available
TIU Category	None	Available	Available
Emotional Stability	Available	Available	Available
Work Achievement	None	Available	Available
Work Tempo	None	Available	Available
accuracy	None	Available	Available
Consistency	None	Available	Available
endurance	None	Available	Available
IQ	Available	Available	Available
IQ Category	None	Available	None
Intelligence	Available	Limited	None
work attitude	Available	Limited	None
IQ. 1	None	Limited	None
Emotional Stability.1	None	Limited	None

Based on the results of the Analysis, it will be divided into two analyses. Analysis based on records focuses on each attribute available in each dataset, namely the attributes 'Seq', 'GPA1', 'GPA2', 'GPA3', 'Emotional Stability,' and 'IQ.' While the second Analysis is an analysis that includes all available column attributes in the 2020 dataset and 2021 dataset or is categorized into Analysis based on features. For comparison, the second Analysis includes all available attributes in each dataset for 2020, and 2021. In this case, the available column attributes are TIU, TIU Category, Emotional Stability, Work Achievement, Work Tempo, Accuracy, Consistency, Endurance, GPA1, GPA2, and GPA3. This is done to enrich the features' availability in this study Architectural Models Based on the architectural design, the research was initiated with the Data Understanding process, which consists of the stages of Data Collection, Describe Data, and Data Validation. Furthermore, the data that has been validated will enter the Data Preparation stage. Data preparation consists of several phases: Data Selection, Data Cleaning, and Data Transformation. In the next stage, Data Integration will be carried out based on the Analysis carried out, namely, Analysis based on records development and Analysis based on features development which produces each dataset. Furthermore, the two datasets will be carried out in the build test scenario stage, the modeling stage. In this condition, split the data with a ratio of 80:20. As much as 80% of the dataset will be used for training (train data) and the remaining 20% for data testing (test data).

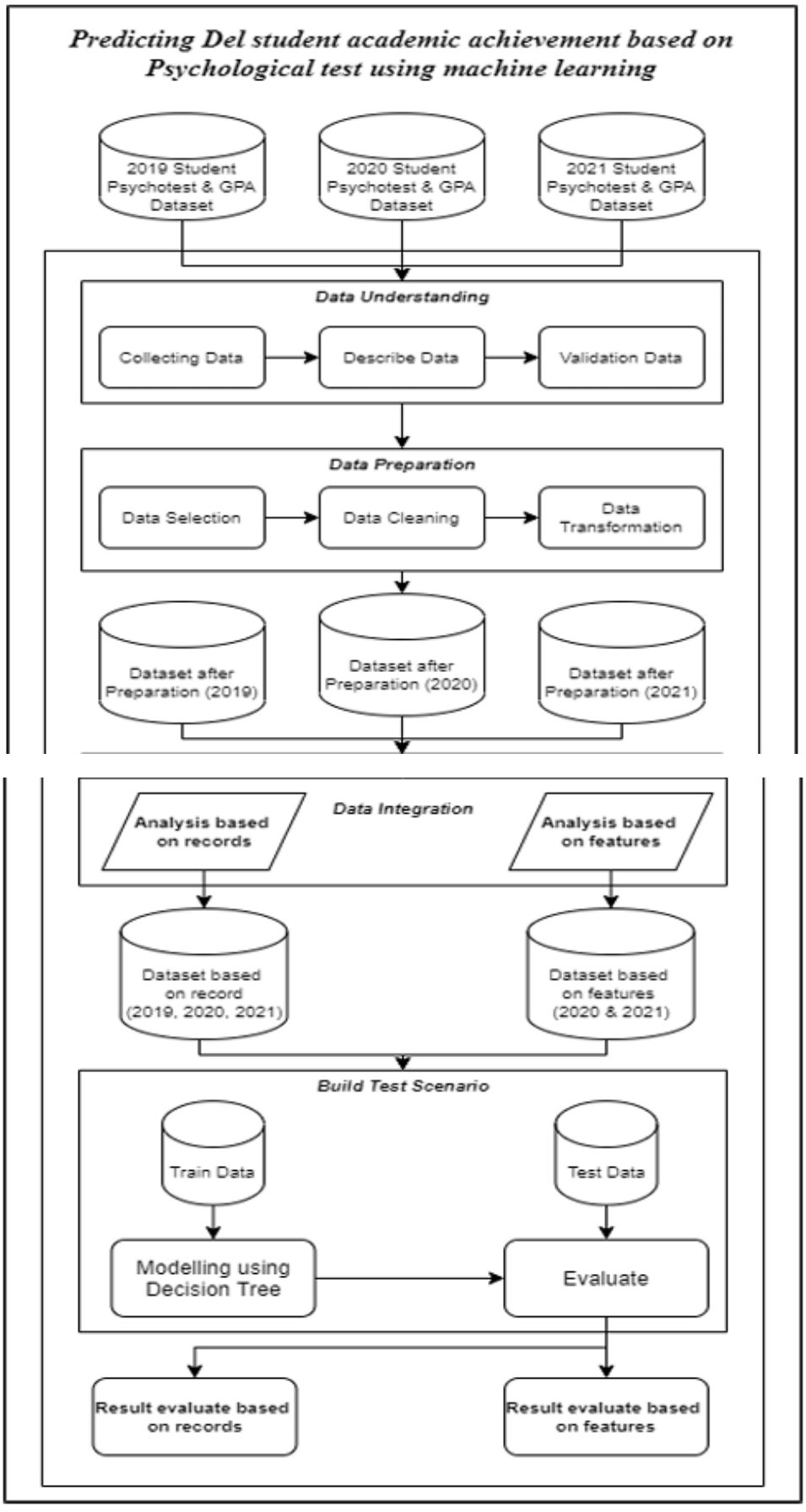
Predicting Del student academic achievement based on Psychological test using machine learning
2019 Student Psychotest & GPA Dataset
2020 Student Psychotest & GPA Dataset
2021 Student Psychotest & GPA Dataset
Data Understanding
Collecting Data
Describe Data
Validation Data
Data Preparation
Data Selection
Data Cleaning
Data Transformation
Dataset after Preparation (2019)
Dataset after Preparation (2020)
Dataset after Preparation (2021)
Data Integration
Analysis based on records
Analysis based on features
Dataset based on record (2019, 2020, 2021)
Dataset based on features (2020 & 2021)
Build Test Scenario
Train Data
Test Data
Modelling using Decision Tree
Evaluate
Result evaluate based on records
Result evaluate based on features

Figure 1: Architectural Design models

Data Correlation

The following section aims to analyze the relationship of each attribute used based on the Analysis based on records and Analysis based on features. This stage aims to explain the relationship conditions for each attribute as the basis for feature selection to be used in model development using heatmap correlation.

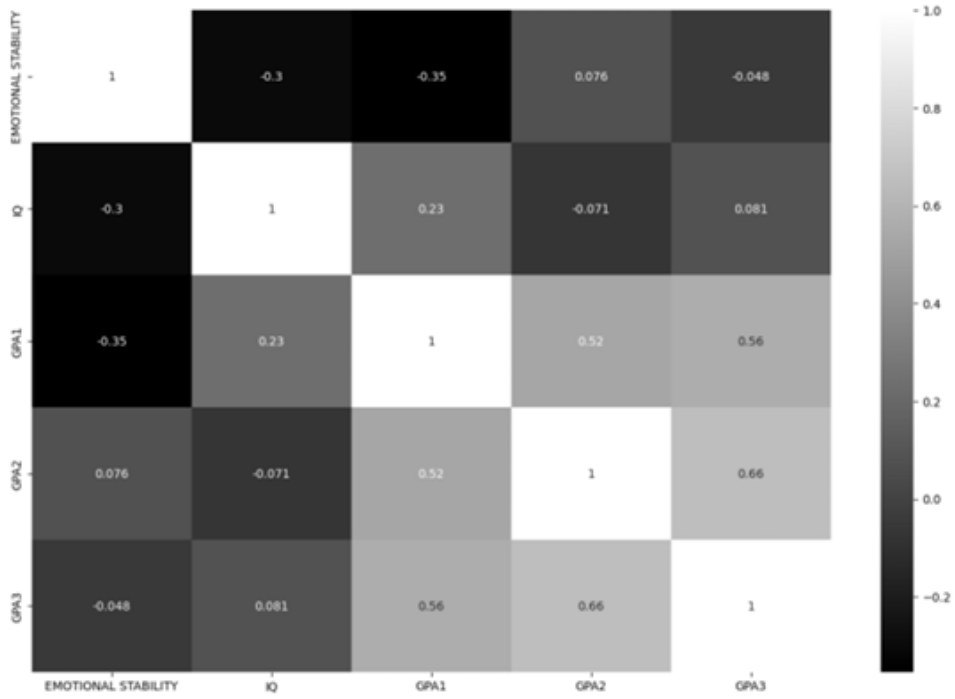

Figure 2: Heatmap correlation based on attributes

The image displays a heatmap correlation based on attributes that match the Analysis based on records showed a very weak correlation between the input feature and the target feature, specifically a correlation of 0.076 between emotional stability and GPA 2, indicating a very weak positive correlation. Additionally, there is a negative correlation of 0.071 between IQ and GPA 2, indicating a very weak negative correlation. The heatmap correlation ranges from zero (0) to one (1).

This section also shows positive values and negative values. A positive value means that the two attributes move in the same direction, while a negative value means the opposite. Zero value indicates that the two attributes have no correlation. On the heatmap, the GPA1, GPA2, and GPA3 attributes are displayed in white, which indicates a strong correlation between each of these attributes. Based on this, we will remove the columns for GPA1 and GPA3 due to the Analysis for feature selection. In addition, GPA2 will be used as a target feature to support the objectives of this study by using GPA as a reference, followed by psychological test scores. As a consideration, GPA1 is not used as a feature selection. It is because students are currently carrying out the adaptation process well in the form of lectures being held. Whereas GPA3 was chosen for data, this means that data for this study has yet to be available, compared to GPA2, which in the future can be used as a research comparison for the 2022 batch dataset if available.

In addition, if you use the GPA1 and GPA3 attributes as input features, this will conflict with research objectives which measure the performance of prospective students based on a combination of psychological tests by comparing GPA results taken by students during their active lecture period or in other words when using GPA1, GPA2, and GPA3 for student performance, there is no need to make predictions involving psychological test attributes. Based on these results, there will be two input features, namely 'EMOTIONAL STABILITY' and 'IQ,' and a target feature, GPA2. As a note, the attributes described in these results will be used in future datasets based on records. All available attributes will be displayed after combining the 2019, 2020, and 2021 datasets (datasets based on records).

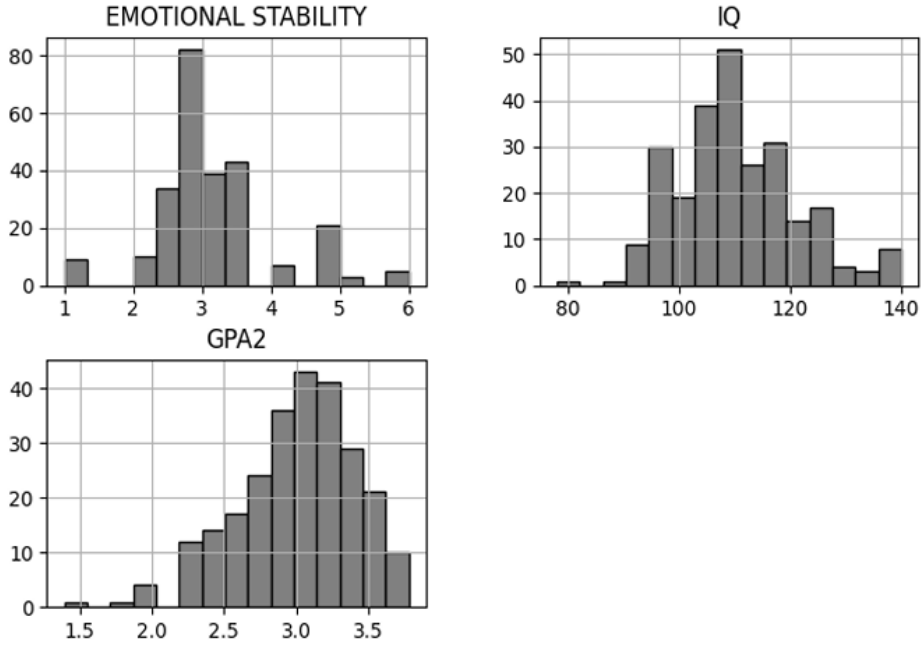

Figure 3: Available attribute of datasets based on records

The applied feature selection will have similar stages as was done in the Analysis based on records by deleted columns on the GPA1 and GPA3 attributes. Heatmap analysis based on feature correlation showed that each psychological test input feature from 'TIU' to 'IQ', showed a very weak correlation with the target feature, namely 'GPA2' which is displayed with a heatmap color that tends to be dark.

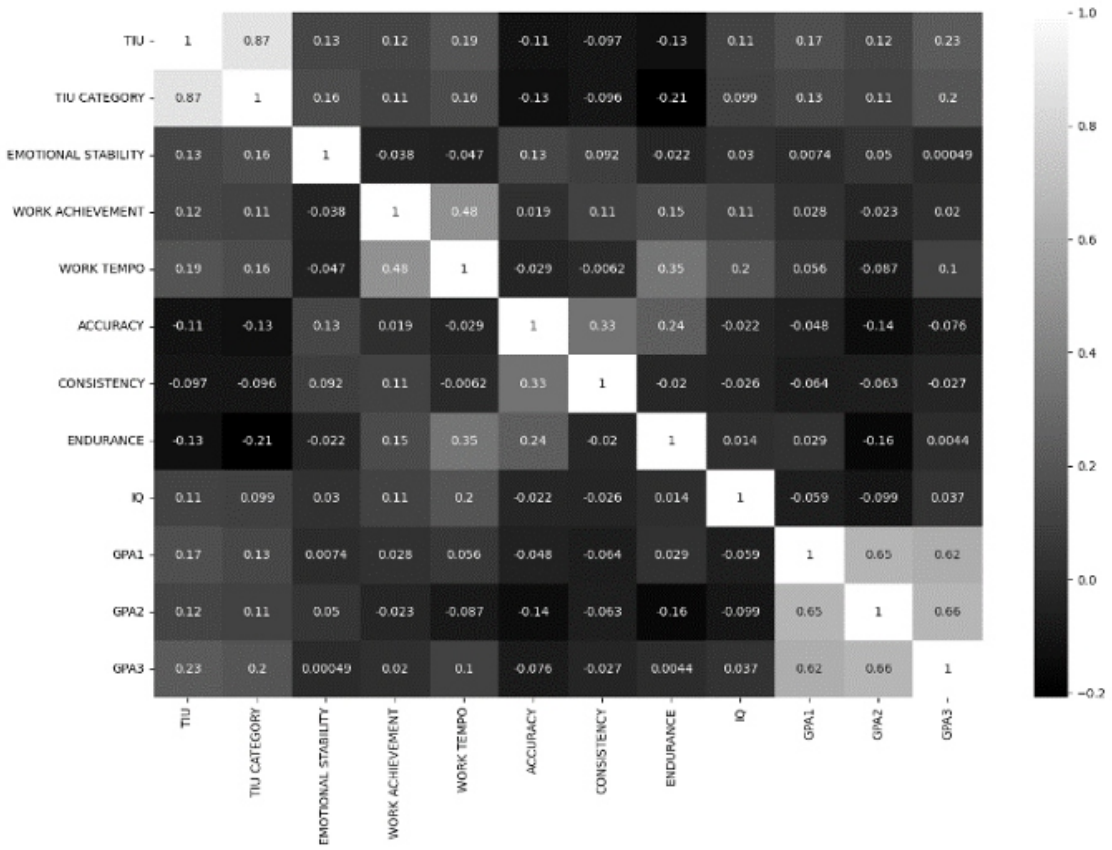

Figure 4: Heatmap correlation based on feature

The correlation heatmap indicates a very weak correlation between the input feature and the target feature, namely a correlation of 0.12 between GPA2 and TIU, a correlation of 0.11 between GPA2 and TIU category, a correlation of 0.05 between GPA2 and emotional stability, and negative correlations of -0.023 with work achievement, -0.14 with accuracy, -0.063 with consistency, -0.16 with endurance, and -0.099 with IQ. All available attributes will be displayed after combining the 2020 and 2021 datasets (datasets based on features).

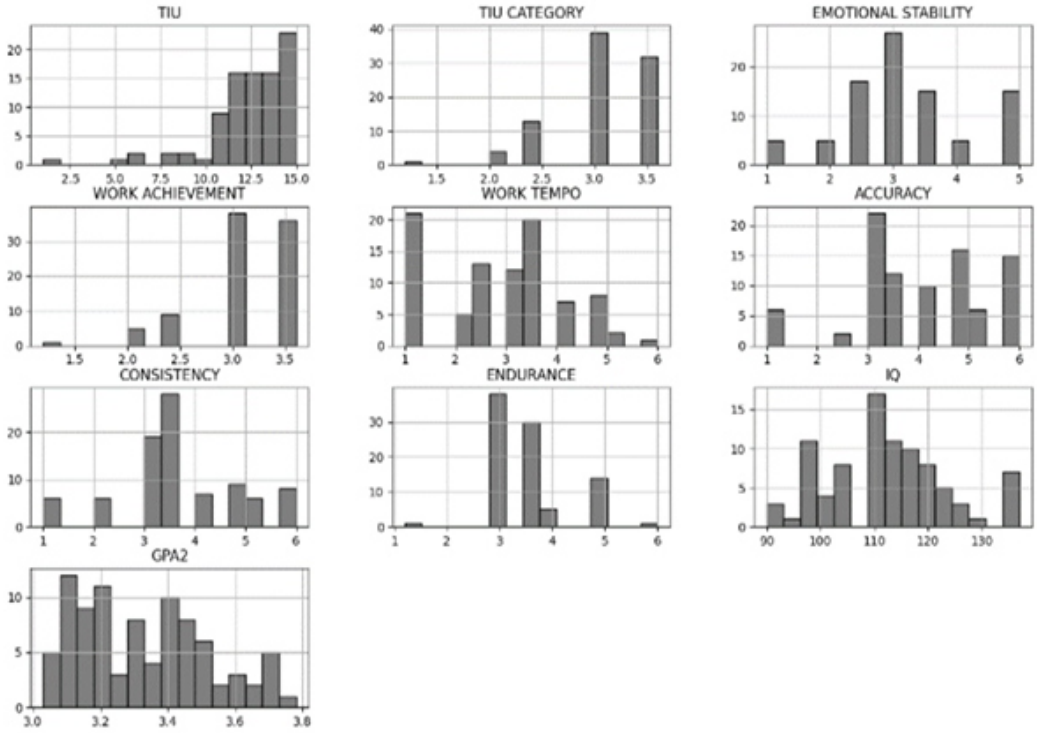

Figure 5: Available attribute of datasets based on features

So based on these conditions, there will be nine input features, namely TIU, TIU Category, Emotional Stability, Work Achievement, Work Tempo, Accuracy, Consistency, Endurance, IQ and GPA2 as target features. These attributes describe the attributes available in the dataset based on features. As an explanation, TIU is General Intelligence Test and TIU Category is labeling in letter form for TIU attributes.

RESULTS

The following will show a bar chart comparing the evaluation results of the first Analysis or Analysis based on records using 253 data with the results of the second Analysis or Analysis based on features using 168 data used to predict the results of GPA2.

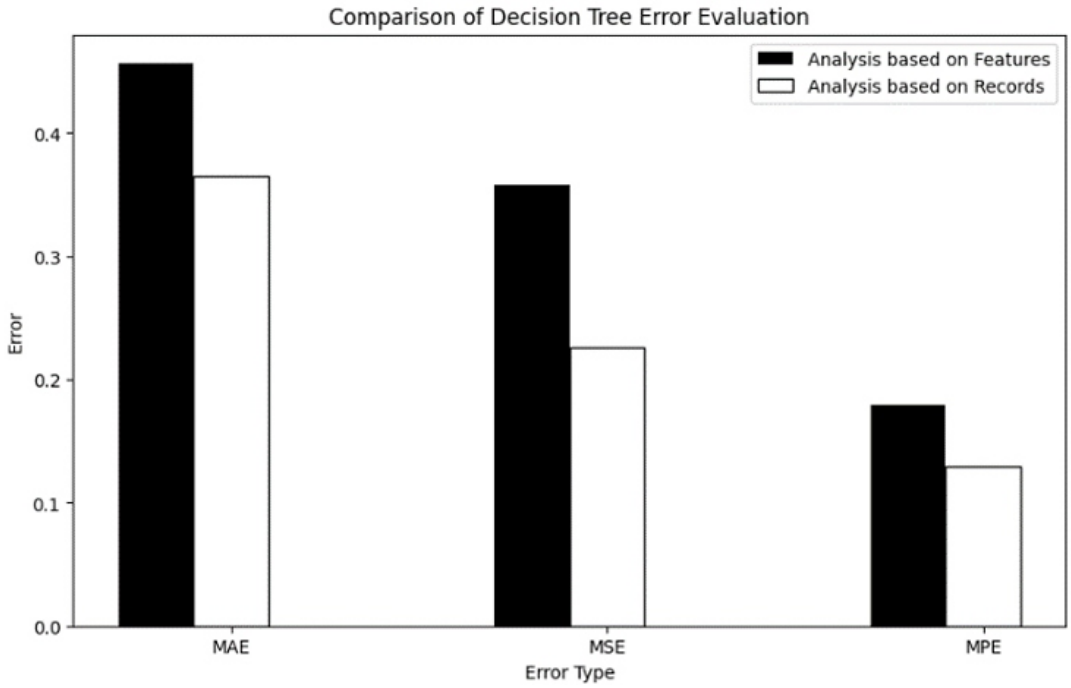

Figure 6: Comparison od Decision Tree Error Evaluation

For special values generated from the evaluation results will be displayed as follows.

Table 3: Evaluation Results

Metrics	Decision Tree based on Records	Decision Tree based on Features
MAE	0.3654	0.4568
MSE	0.2263	0.3577
MPE	0.1296	0.1797

Based on the evaluation above, the results show that decision tree algorithms tend to produce low error evaluations, or models with better performance are used when using more features during model training and testing. In more detail, the lowest error evaluation results can be measured through MAE. This aims to compare the smaller output of the two analyses carried out. Thus the development of the best model that produces the lowest error and by the research title Decision Tree Algorithm for Predicting Student Performance Based on Psychological Test is Analysis based on features.

CONCLUSION

Based on the research analysis that has been done, namely Analysis based on features and Analysis based on records, a comparison is made between the evaluation error of the decision tree algorithm. It can be concluded that the best performance is analysis based on records. The Analysis based on records showed a very weak correlation between the input feature and the target feature, specifically a correlation of 0.076 between emotional stability and GPA 2, indicating a very weak positive correlation. Additionally, there is a negative correlation of -0.071 between IQ and GPA 2, indicating a very weak negative correlation. This is as shown in the correlation heatmap, which shows the level of correlation

for each attribute is still classified as very weak correlation. Meanwhile the evaluation results show a relatively low and good relative error, with MAE of 0.3654, MSE of 0.2263, and MPE of 0.1296. Even though the development of machine learning models has been successfully carried out in research, certain psychological test aspects are not a guarantee in determining each student's performance. Therefore, it is important to recognize that assessing student achievement should not solely focus on psychological tests or academic exams but should also consider aspects such as motivation, personality, interests, social skills, and creativity. Taking a holistic and comprehensive approach to evaluate student achievement can provide a more complete and accurate picture of their abilities and potential.

REFERENCES

[1] H. Altabrawee, O. A. J. Ali, and S. Q. Ajmi, "Predicting Students' Performance Using Machine Learning Techniques," J. Univ. BABYLON Pure Appl. Sci., vol. 27, no. 1, pp. 194205, 2019, doi: 10.29196/jubpas.v27i1.2108.

[2] I. Baron, H. Agustina, and Melania, "Journal of Management and Marketing Review The Role of Psychological Testing As an Effort to Improve Employee Competency," J. Manag. Mark. Rev., vol. 5, no. 1, pp. 1–15, 2020, [Online]. Available: https://doi.org/10.35609/jmmr.2020.5.1.

[3] E. Tanuar, Y. Heryadi, Lukas, B. S. Abbas, and F. L. Gaol, "Using Machine Learning Techniques to Earlier Predict Student's Performance," 1st 2018 Indones. Assoc. Pattern Recognit. Int. Conf. Ina. 2018 - Proc., pp. 85–89, 2019, doi: 10.1109/INAPR.2018.8626856.

[4] W. R. Russell, "Psychological Tests in Neurology," Bmj, vol. 1, no. 5330, pp. 602–603, 1963, doi: 10.1136/bmj.1.5330.602b.

[5] M. S. Acharya, A. Armaan, and A. S. Antony, "A comparison of regression models for prediction of graduate admissions," ICCIDS 2019 - 2nd Int. Conf. Comput. Intell. Data Sci. Proc., pp. 1–5, 2019, doi: 10.1109/ICCIDS.2019.8862140.

[6] H. Dabiri, V. Farhangi, M. J. Moradi, M. Zadehmohamad, and M. Karakouzian, "Applications of Decision Tree and Random Forest as Tree-Based Machine Learning Techniques for Analyzing the Ultimate Strain of Spliced and Non-Spliced Reinforcement Bars," Appl. Sci., vol. 12, no. 10, pp. 1–13, 2022, doi: 10.3390/app12104851.

[7] R. Kumar, P. Kumar, and Y. Kumar, "Time Series Data Prediction using IoT and Machine Learning Technique," Procedia Comput. Sci., vol. 167, no. 2019, pp. 373–381, 2020, doi: 10.1016/j.procs.2020.03.240.

[8] A. Taufiqurrahman, A. G. Putrada, and F. Dawani, "Decision Tree Regression with AdaBoost Ensemble Learning for Water Temperature Forecasting in Aquaponic Ecosystem," 6th Int. Conf. Interact. Digit. Media, ICIDM 2020, no. Icidm, 2020, doi: 10.1109/ICIDM51048.2020.9339669.

[9] L. He, S. Diego, R. A. Levine, and S. Diego, "Random Forest as a Predictive Analytics Alternative to Regression in Institutional Research," vol. 23, no. 1, 2018.

[10] M. G. Uddin and M. Uddin, "E-Government Development & Digital Economy: Relationship," Am. Econ. Soc. Rev., vol. 6, no. 1, pp. 39–54, 2020, doi: 10.46281/aesr.v6i1.580.

Machine Learning Fusion Algorithm using for Forecasting Thyroid Disease

K. P. Manikandan 1*, B. Anusha 2, S. Girija 3, K. Harshitha 4, M. Keerthana Royal 5

1 Assistant Professor, Department of CSE, Madanapalle Institute of Technology & Science, Madanapalle, Andhra Pradesh, India

2, 3, 4, 5 Department of CSE, Madanapalle Institute of Technology & Science, Madanapalle, Andhra Pradesh, India

ABSTRACT

This paper proposes several feature selection and classification procedures for thyroid ailment diagnosis, which is one of the most critical classification issues. Two Thyroid disease refers to a set of disorders affecting the thyroid gland, which produces thyroid hormones. Hormones are in charge of controlling the pace of metabolism in the body. Hyperthyroidism and hypothyroidism are two types of thyroid diseases. They are classified. Thyroid disease is a challenging issue to resolve. The process of extracting or choosing a group of features is an important challenge in the field of pattern recognition. This is a step in the pre-processing process. As an example, consider the word sequence. The words "sequence backward selection" and "ahead selection" are used interchangeably. Two well-known heuristic approaches are utilized for feature extraction. selection. Genetics is a science. In the health system, where there is a huge amount of data and information to manage, machine learning algorithms are essential for dealing with data. Our study on thyroid disease employed machine learning techniques. With the aim of classifying thyroid disease into three groups—hyperthyroidism, hypothyroidism, and normal—we conducted this study using data from Iraqi individuals, some of whom have hyperthyroidism and others who have hypothyroidism.

Keywords : AdaBoost, Decision Tree, Support Vector Machines and XgBoost.

INTRODUCTION

Thyroid illness, a branch of endocrinology, is one of the least understood and recognized diseases [1]. Diabetes is the most common endocrine illness, followed by thyroid issues, according to the World Health Organization [2]. Hypothyroidism and hyperfunctioning hyperthyroidism afflict 2% and 1% of the population, respectively [3]. Men are involved in around one-tenth of all instances. Thyroid dysfunction caused by pituitary gland failure or hypothalamic dysfunction can result in hyper- and hypothyroidism [4].

Goitres, or active thyroid nodules, can occur at a rate of up to 15% in areas where dietary iodine levels are low. Furthermore, a variety of malignancies can form in the thyroid gland [5], making it a potentially dangerous location [6]. Endocrine glands, particularly the thyroid, create and distribute hormones throughout the body [7]. It is placed in the front center of the body. Hormones produced by the thyroid gland govern physiological fluid balance, digestion, and other processes [8] [9]. T3 (triiodothyronine), T4 (thyroid hormone), and TSH (thyroid stimulating hormone) can all be used to treat the thyroid gland [10]. Thyroid problems are classified as hypothyroidism or hyperthyroidism [11]. Data mining is a semi-automated approach for discovering patterns in vast volumes of data [12]. Hyperthyroidism is characterized by an excess of thyroid hormones produced by the thyroid gland [13]. Hyperthyroidism is caused by an increase in thyroid hormone levels [14]. Trembling, dry skin, increased sensitivity to heat,

thinning hair, weight loss, an elevated heart rate, high blood pressure, excessive perspiration, neck enlargement, anxiety [15], shorter menstrual cycles, irregular stomach motions, and an expanded neck are some of the symptoms. Hypothyroidism causes the thyroid gland to become underactive [16] [17]. Hypothyroidism is caused by a decrease in thyroid hormone production [18]. In medical words, hypo denotes inadequate or less. Inflammation and thyroid gland injury are the two most prevalent causes of hypothyroidism [19]. Obesity, low heart rate, increased sensitivity to heat, neck swelling, dry skin, numb hands, hair difficulties, heavy menstrual periods, and digestive disorders are also symptoms [20].

PROPOSED MODEL

Several machine learning methods for classifying thyroid problems have been developed, but none fully address the issue of misdiagnosis. Furthermore, comparable research has presented methods to analyses this illness categorization, but they typically overlook the data's magnitude and heterogeneity. As a result, we recommend Support Vector Machines, XGBoost, Decision Trees, and AdaBoost. Performing machine learning-based classifier testing.

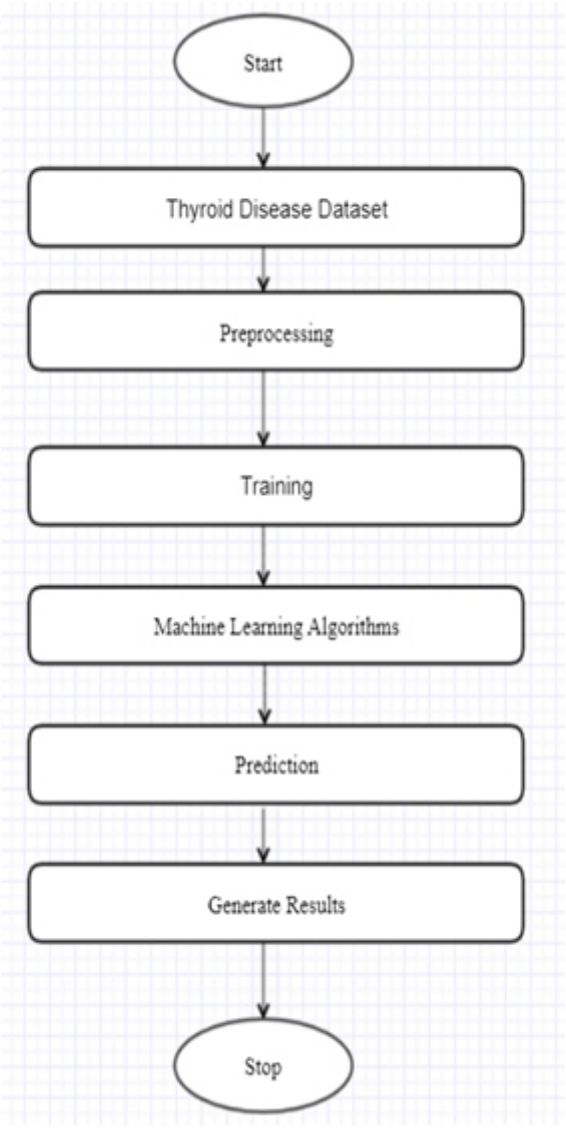

Figure 1. Block diagram of Thyroid Disease

Architecture

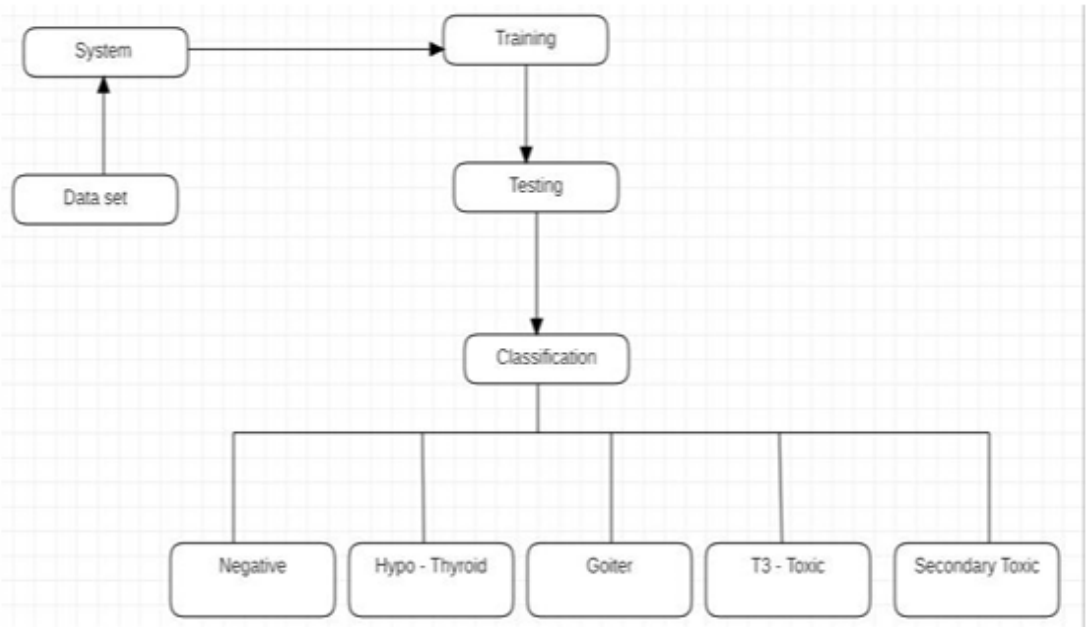

Figure 2. Architecture of Thyroid Disease

Decision Tree

Each core node in a flowchart-like structure represents a feature (or attribute), a branch represents a decision rule, and each leaf node represents the conclusion. A decision tree's root node is the node at the top. It gains the power to divide based on how important particular features are. This type of tree division is known as recursive partitioning. This decision-making method is comparable to a flowchart. It is a flowchart-like image that perfectly represents human level thinking.

As a consequence, decision trees are straightforward to comprehend and analyses.

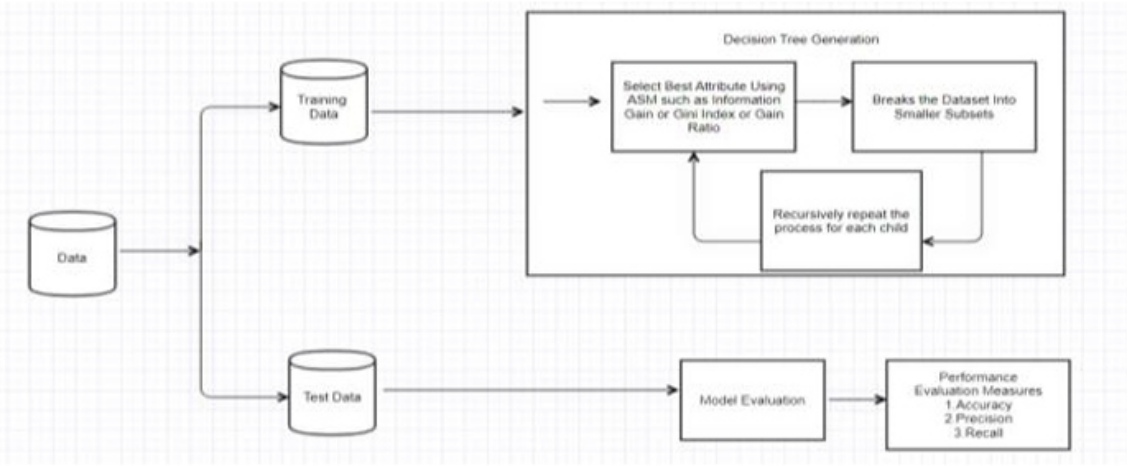

Figure 3. Generating Dataset using Decision Tree

XG Boost

The acronym XGBoost stands for Extreme Gradient Boosting. XGBoost is a distributed gradient boosting library that is exceptionally efficient, versatile, and portable. To develop machine learning algorithms, use the Gradient Boosting framework. allows parallel tree boosting to perform a variety of data science queries quickly.

Adaboost Classifier

In 1996, Yoav Freund and Robert Schapire introduced Ada-Boost, also known as adaptable boosting, as an ensemble boosting classifier. To improve accuracy, combine numerous classifiers. AdaBoost is a method for creating iterative ensembles. The AdaBoost classifier constructs a strong classifier that is extremely accurate by merging a number of weak classifiers. Adaboost's central tenet is to train data samples and create classifier weights at each iteration to ensure accurate prediction of anomalous events. A simple classifier is any machine learning method that gives weights to the training data. Adaboost must meet two requirements:

1. To train the classifier interactively, a variety of weighted training cases should be employed.
2. It attempts to offer a suitable match for these occurrences in each iteration by categorizing training mistake. It works in the following manner:

i. Adaboost begins by selecting a random member of the education subset.

ii. It trains the AdaBoost machine learning model repeatedly by picking the training set based on the accuracy of the previous training.

iii. It provides erroneously classified data additional weight, increasing the possibility that these observations will be categorized in the next cycle.

iv. Weight is assigned to the trained classifier in each iteration based on its accuracy. The classifier with the highest accuracy will be given more weight.

v. This method is used repeatedly until the task is completed.

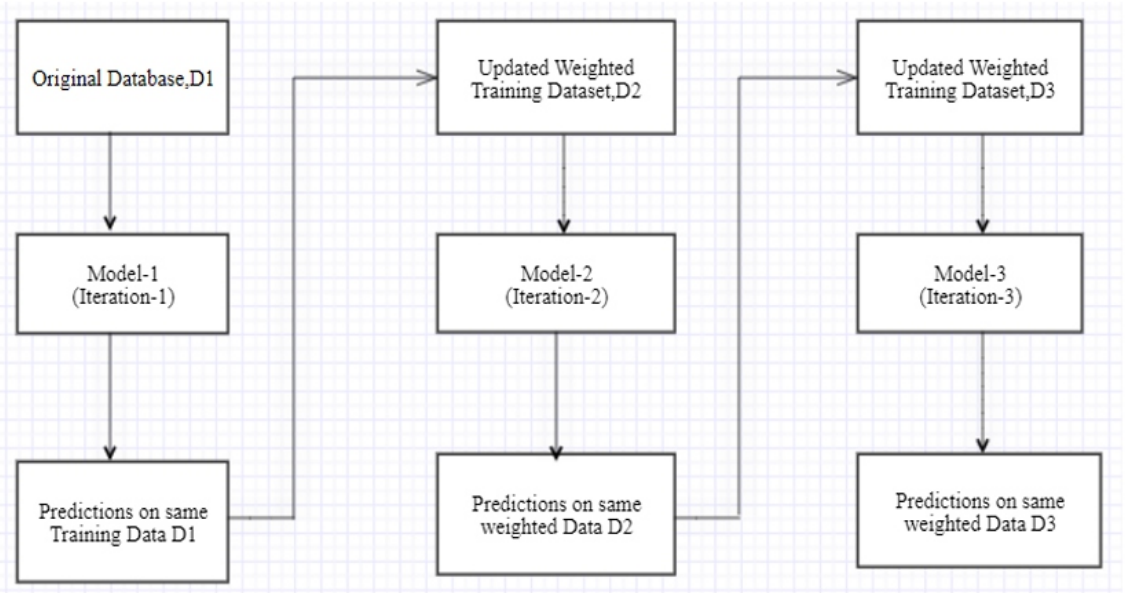

Figure 4. Adaboost Classifier

Support Vector Machine

Support Vector Machines are commonly considered of as a classification approach, despite the fact that

they can address both classification and regression issues. It easily handles a large number of continuous and categorical variables. To discriminate between classes, SVM creates a hyperplane in multidimensional space. To reduce inaccuracy, SVM leverages

SVM's basic principle is to determine the maximum marginal hyperplane (MMH) that optimally divides the dataset into classes. repeatedly created optimum hyperplanes. The classifier with the highest accuracy will be given more weight.

In the following steps, maximize the marginal hyperplane:

1. Use hyperplanes to get the best feasible class separation. The left-hand figure depicts black, blue, and orange hyperplanes. While the black successfully identifies the two groups, the blue and orange exhibit more classification mistakes in this circumstance.
2. Select the hyperplane in the right image that is farthest distant from the two closest data points.

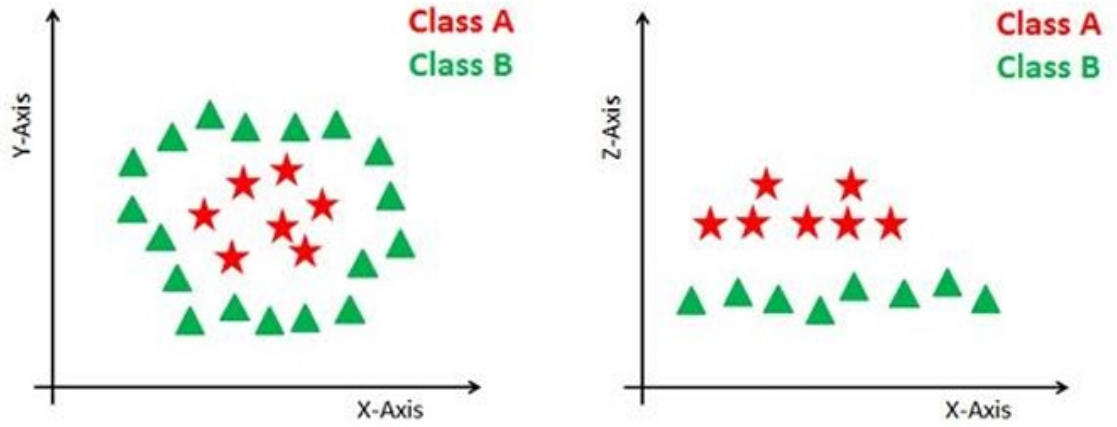

Figure 5. Support Vector Machine Classifier

EXPERIMENTAL RESULTS AND DISCOVERY

The thyroid dataset shown below was utilized to analyses the experimental results. Four algorithms were used in this study: Decision tree, XGBoost, AdaBoost, and Support vector machine.

Dataset

Table 1. Thyroid Disease Dataset

age	Sex	goitre	psych	on_thyroxine
35	F	f	f	f
63	M	f	f	f
25	F	f	f	f
53	F	f	f	f
92	F	f	f	f
67	M	f	f	f
60	F	f	f	f
60	F	f	f	f
48	F	f	f	f

27	F	f	f	f
73	F	f	f	f
19	M	f	f	f
72	F	f	f	f
16	M	f	f	f
54	F	f	f	f
39	F	f	f	t
38	M	f	t	f

age	Sex	goitre	psych	on_thyroxine
33	F	f	f	f
45	F	f	f	f
54	F	f	f	f
21	F	t	f	f
19	M	f	f	f
51	F	F	f	F
63	F	F	f	F
51	F	F	f	F
64	M	F	f	F
19	F	F	f	F
40	F	F	f	F
54	F	F	f	F
19	F	F	t	F

33	F	F	f	F
71	M	F	f	F
49	F	F	f	F
79	F	F	f	F
21	F	F	f	F
20	F	F	f	F
20	F	T	f	F
79	M	F	f	F
64	F	F	f	T
59	F	F	f	F

Figure 6. Thyroid disease Classifier

Figure 7. Information about Thyroid Disease

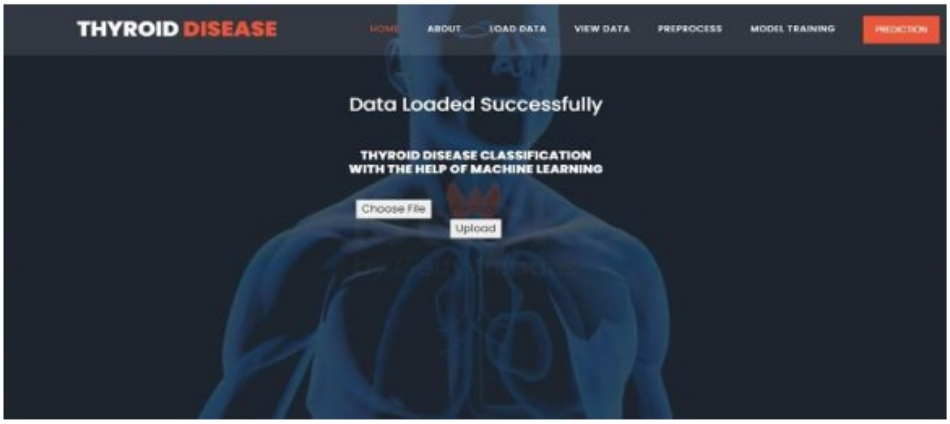

Figure 8. Classifier Data loaded successfully

Figure 9. Classifier dataset displayed

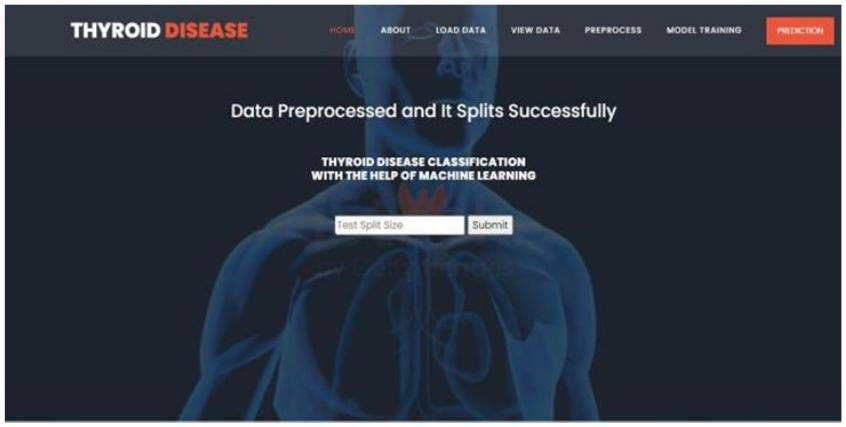

Figure 10. Classifier data splits successfully

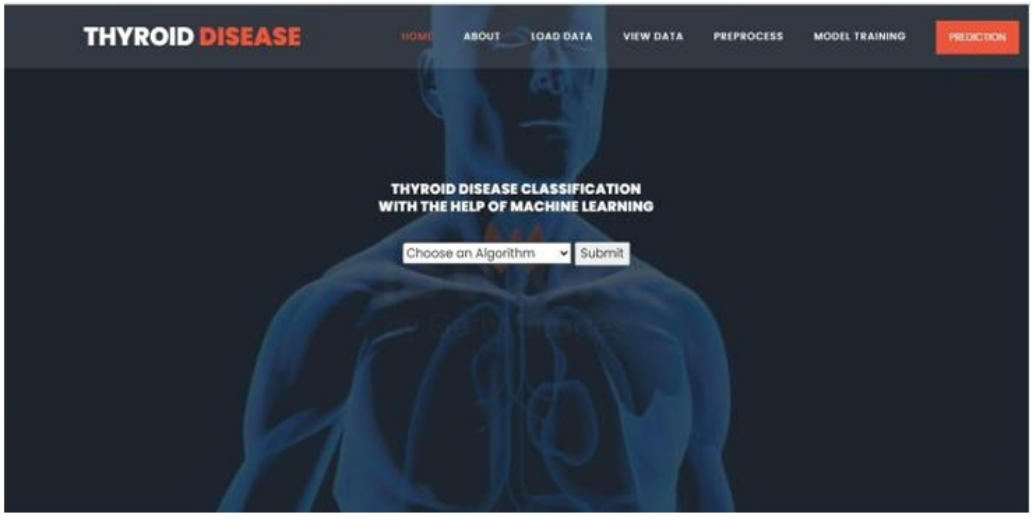

Figure 11. Classifier choosing Machine Learning algorithms

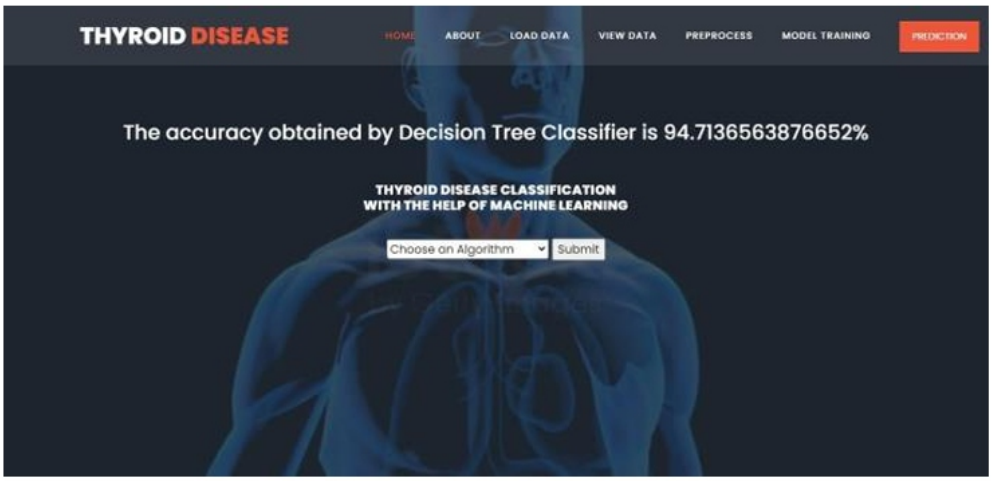

Figure 12. Accuracy obtained by the Classifier

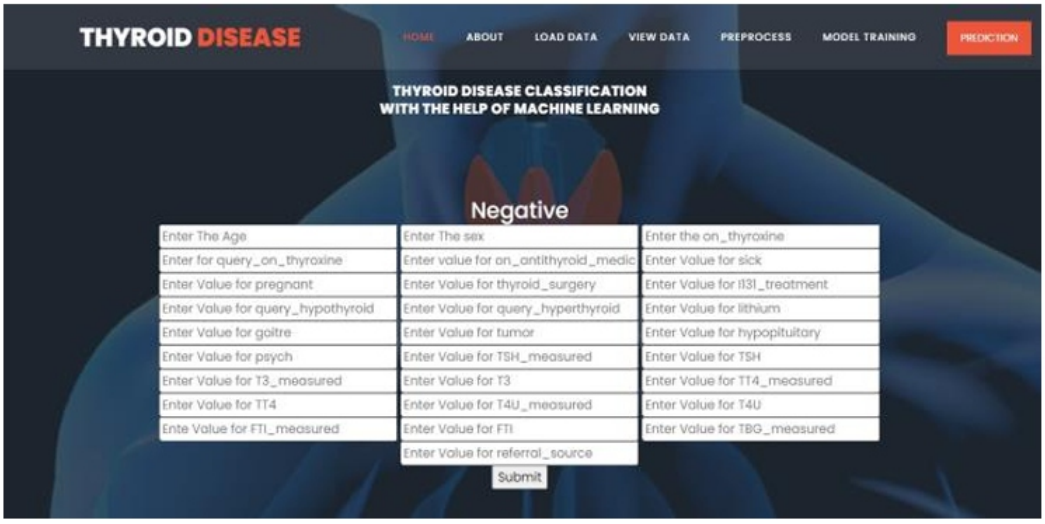

Figure 13. Thyroid Disease Classification with the help of Machine Learning

CONCLUSION

Thyroid disease is one of the ailments that is affecting the worldwide population and is getting more common. Based on medical reports that demonstrate a serious imbalance in thyroid illness, our study investigates the division of thyroid disease into hyperthyroidism and hypothyroidism. The sickness was classified using an algorithm. Machine learning has created two models with promising outcomes using a variety of techniques. All of the characteristics in the first model, which has 16 inputs and 1 output, are captured, and the Ada-boost approach outperforms the other algorithms with a score of 97.35. According to previous study, the second embodiment lacks the following properties. The characteristics 1-query_thyroxine was removed. 2. search for "hypothyroid," 3. search for "hyperthyroid".

REFERENCES

[1] O. Senashova and M. Samuels, "Diagnosis and management of nodular thyroid disease," Vascular and Interventional Radiology, vol. 25, no. 2, article 100816, 2022.

[2] X. Zhang, V. C. Lee, J. Rong, J. C. Lee, and F. Liu, "Deep convolutional neural networks in thyroid disease detection: a multi-classification comparison by ultrasonography and computed tomography," Computer Methods and Programs in Biomedicine, vol. 220, article 106823, 2022.

[3] M. Kang, T. S. Wang, T. W. Yen, K. Doffek, D. B. Evans, and S. Dream, "The clinical utility of preoperative thyroglobulin for surgical decision making in thyroid disease," Journal of Surgical Research, vol. 270, pp. 230–235, 2022.

[4] Chaubey, D. Bisen, S. Arjaria, and V. Yadav, "Thyroid disease prediction using machine learning approaches," National Academy Science Letters, vol. 3, pp. 128–133, 2021. disease

[5] Dewangan, A. Shrivas, and P. Kumar, "Classification of thyroid with feature selection technique," International Journal of Engineering & Technology, vol. 2, no. 3, pp. 128–133, 2021.

[6] K. Shankar, S. Lakshmanaprabu, D. Gupta, A. Maseleno, and V. Albuquerque, "Optimal feature-based multi-kernel SVM approach for thyroid disease classification," The Journal of Supercomputing, vol. 28, no. 76, pp. 1128–1143, 2020.

[7] P. Poudel, A. Illanes, M. Sadeghi, and M. Friebe, "Patch based texture classification of thyroid ultrasound images using convolutional neural network," in 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 5828–5831, Berlin, Germany, 2019.

[8] M. Guo and D. Yongzhao, "Classification of thyroid ultrasound standard plane images using ResNet-18 networks," in 2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID), pp. 324–328, Xiamen, China, 2019.

[9] Ma, J. Guan, W. Zhao, and C. Wang, "An efficient diagnosis system for Thyroid disease based on enhanced Kernelized Extreme Learning Machine Approach," in International Conference on Cognitive Computing, pp. 86–101, Cham, 2018.

[10] Aswathi and A. Antony, "An intelligent system for thyroid disease classification and diagnosis," in 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), pp. 1261–1264, Coimbatore, India, 2018.

[11] K. Chandel, S. Veenita Kunwar, T. C. Sabitha, and S. Mukherjee, "A comparative study on thyroid disease detection using K-nearest neighbor and naïve Bayes classification techniques," CSI Transactions on ICT, vol. 4, no. 2-4, pp. 313–319, 2018.

[12] K. Shankar, S. Lakshmanaprabu, D. Gupta, A. Maseleno, and V. Albuquerque, "Optimal feature-based multi-kernel SVM approach for thyroid disease classification," The Journal of Supercomputing, vol. 28, no. 76, pp. 1128–1143, 2017.

[13] Dr. Srinivasan B, Pavya K "Diagnosis of Thyroid Disease: A Study" International Research Journal of Engineering and Technology Volume: 03 Issue: 11 | Nov – 2016.

[14] Khushboo Taneja, Parveen Sehgal, Prerana "Predictive Data Mining for Diagnosis of Thyroid Disease using Neural Network" International Journal of Research in Management, Science & Technology (E-ISSN: 2321- 3264) Vol. 3, No. 2, April 2016.

[15] Kouroua, K., Exarchosa, T.P. Exarchosa, K.P., Karamouzisc, M.V. and Fotiadisa, D.I. (2015) Machine learning applications in cancer prognosis and prediction, Computational and Structural Biotechnology Journal, Vol. 13, Pp.8–17.

[16] S. Prerana and K. Taneja, "Predictive data mining for diagnosis of thyroid disease using neural network," International Journal of Research in Management, Science & Technology, vol. 3, no. 2, pp. 75–80, 2015.

[17] Travis B Murdoch and Allan S Detsky. The inevitable application of big data to health care. Jama,

309(13):13511352, 2013.
[18] Azar, a.T, Hassanien, A.E. and Kim, T. Expert system based on neural fuzzy rules for thyroid diseases diagnosis, Computer Science, Artificial Intelligence, arXiv:1403.0522, Pp. 1-12,2012.
[19] Shukla, A. & Kaur, P. (2009). Diagnosis of thyroid disorders using artificial neural networks, IEEE International Advance computing Conference (IACC 2009)– Patiala, India, pp 1016-1020.
[20] Keles, A. ESTDD: Expert system for thyroid diseases diagnosis, Expert Syst Appl., Vol. 34, No.1, Pp.242246,2008.

Fake Review Detection using Machine Learning

Dr. M Gayathri 1, Y.S.N Siva Teja 2*, K.Ajay Sharma3
1 Assistant Professor, Department of CSE, SCSVMV University, Kanchipuram, India
2, 3 Student, SCSVMV University, Kanchipuram, India

A B S T R A C T

Online reviews have become increasingly important in the world of e-commerce, serving as a powerful tool to establish a business's reputation and attract new customers. However, the rise of fake reviews has become a growing concern as they can skew the reputation of a business and deceive potential customers. As a result, detecting fake reviews has become a key area of research in recent years. This project proposes a machine learning-based approach to detect fake reviews. The method utilizes various feature engineering techniques to extract different behavioural characteristics of reviewers, such as the length of reviews and the frequency of review submissions. These characteristics are then used to train different algorithms, including K-Nearest Neighbors (KNN), Random Forest, and Support Vector Machine (SVM), to classify reviews as either genuine or fake. The proposed technique was evaluated using a real dataset extracted from the internet, and the results showed that SVM outperformed the other classifiers in terms of accuracy. This suggests that SVM is a powerful algorithm for distinguishing between genuine and fake reviews. However, the study also suggests that there is potential to improve the performance of the model by integrating more behavioural characteristics of reviewers, such as how frequently they do reviews and how long it takes them to complete reviews. In conclusion, this project highlights the importance of detecting fake reviews and proposes a machine learning-based approach to achieve this. The study shows that SVM is a powerful algorithm for this task, but there is potential for further improvement by incorporating more reviewer behavioural characteristics. The findings of this research have practical implications for businesses, consumers, and researchers in the field of e-commerce.

Keywords : Customers, E-Commerce, Fake Reviews, Machine learning.

INTRODUCTION

In today's digital age, reviews are the primary source of information for customers when making purchasing decisions. Whether it's a product or service, customers rely on the feedback and opinions of others to gauge its value, features, and ratings. As a result, reviews have become a crucial source of authentic data for most people in online services [1]. However, the presence of fake reviews that aim to mislead customers has become a significant concern. Detecting fake reviews has thus become a crucial and active research area.

Machine learning techniques can play a crucial role in detecting fake reviews from online content. One of the web mining tasks, content mining, involves extracting useful information using various machine learning algorithms. Opinion mining is a common example of content mining, where sentiment analysis techniques are used to analyze the sentiment of the text, positive or negative [2]. Detecting fake reviews, however, requires building features that go beyond the content itself, such as the reviewer's style, review time/date, or other attributes. Thus, successful fake review detection lies in the development of meaningful features that can accurately identify fake reviews [3].

EXISTING SYSTEM

Machine learning is becoming more prevalent, with classical and machine learning approaches used in computer science. This section discusses relevant research on detecting fake reviews and how machine learning techniques outperform conventional ones [4]. However, the current methodology has following Disadvantages.

Disadvantages

Low Accuracy.
High Time and space complexity.
Highly inefficient in terms of memory.
Requires skilled persons for operation.

PROPOSED SYSTEM

To address the limitations of the existing system, we propose a new application that leverages machine learning techniques and a Python-based environment. The goal of this project is to provide a reliable, quick, and accurate mechanism for detecting fake reviews. We used robust methods and various machine learning algorithms to develop this system, and it has the potential for future updates based on the evolving requirements of fake review detection [5].

In summary, our proposed system is expected to provide significant benefits in detecting fake reviews and contribute to the field's growth. It aims to overcome the limitations of the existing system, such as low accuracy and high time complexity, and provide an efficient and trustworthy mechanism for identifying fake reviews.

Algorithm

Support Vector Machine (SVM) Classification

Support Vector Machine (SVM) is a popular algorithm used in machine learning for classification and regression problems. The primary use of SVM is for classification problems, where the goal is to classify data points into different categories based on a set of features. SVM aims to create the best line or decision boundary that can effectively separate the data points into different classes [6]. The decision boundary created by SVM is known as a hyperplane, and the algorithm selects extreme points or vectors known as support vectors to create this hyperplane.

The support vectors are data points that are closest to the decision boundary and play a significant role in the creation of the hyperplane. SVM is widely used for various applications such as face detection, image classification, text categorization, and more [7]. For instance, in image classification, SVM can be used to identify whether an image contains a specific object or not. SVM can also be used in text classification to categorize text into different categories such as positive or negative sentiment. One of the advantages of SVM is that it can handle high-dimensional data with ease. SVM is also effective when the number of features is greater than the number of data points. SVM can also handle non-linear decision boundaries by using kernel functions. In conclusion, SVM is a powerful algorithm for classification problems and has several applications in various fields. Its ability to handle high-dimensional data and non-linear decision boundaries makes it a popular choice for many machine learning tasks [8].

Architecture

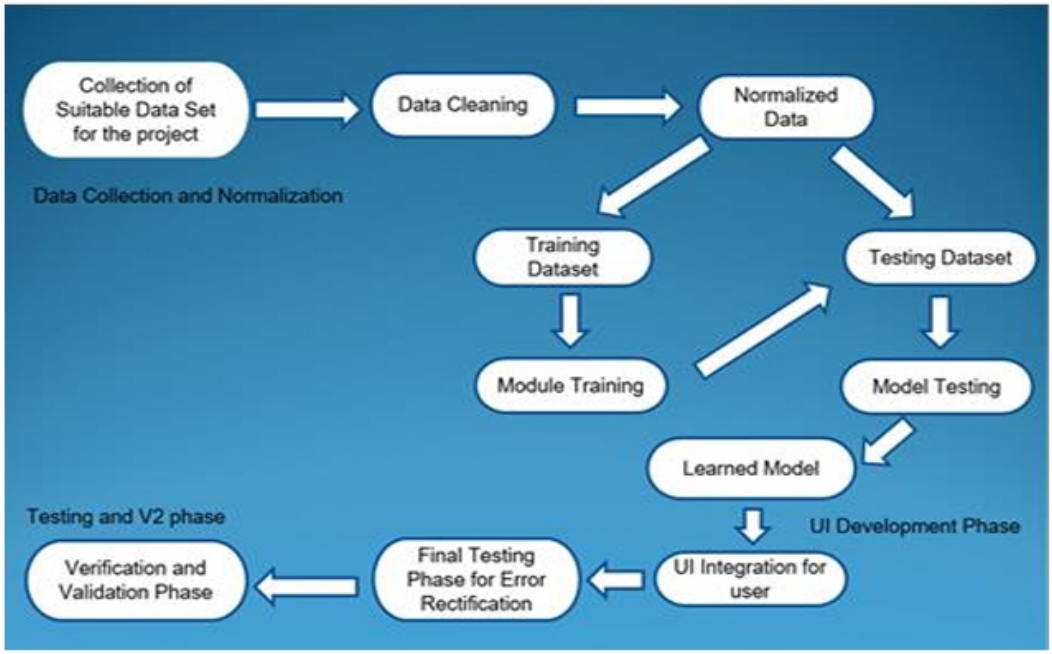

Figure 1. System Architecture Diagram

The above system architecture diagram (Figure 1) depicts the method for developing the project from initial state to the final stage. The stages of development are described clearly.

MODULE DESCRIPTION

In this, we have created a machine learning model that is effective at predicting whether or not an internet review is false. The fundamental principle utilized to identify reviews as being fake is that they should be produced unfairly by machine [6]. The review is regarded as legitimate and authentic if it was written by hand. And we also developed a sample commerce website incorporated with this machine learning model for user understanding and for producing real time results [9][10].

RESULTS

Table 1. Accuracy of Different Algorithms

S.No	Algorithm Name	Accuracy
1	K Nearest Neighbours Algorithm	57.67%
2	Random Forest Algorithm	83.53%
3	Support Vector Machine Algorithm	87.72%

The above table (Table 1) provides the accuracy of different algorithms. This accuracy is calculated with the help of testing data set in Testing Phase.

```
In [45]: print('Performance of various ML models:')
         print('\n')
         print('K Nearest Neighbors Prediction Accuracy:',str(np.round(accuracy_score(label_test,knn_pred)*100,2)) + '%')
         print('Random Forests Classifier Prediction Accuracy:',str(np.round(accuracy_score(label_test,rfc_pred)*100,2)) + '%')
         print('Support Vector Machines Prediction Accuracy:',str(np.round(accuracy_score(label_test,svc_pred)*100,2)) + '%')

         Performance of various ML models:


         K Nearest Neighbors Prediction Accuracy: 57.67%
         Random Forests Classifier Prediction Accuracy: 83.53%
         Support Vector Machines Prediction Accuracy: 87.72%
```

Figure 2. Results

Description of Results

From the above results (Figure 2), we can clearly see that Support Vector Machine (SVM) performs clearly in classifying the reviews as OR or CG followed by Random forest classifier and KNN is averagely performing by giving only 57% accuracy. Thus, we can conclude that Support Vector Machine (SVM) with an accuracy of 87.72% is a clear winner in detecting fake reviews.

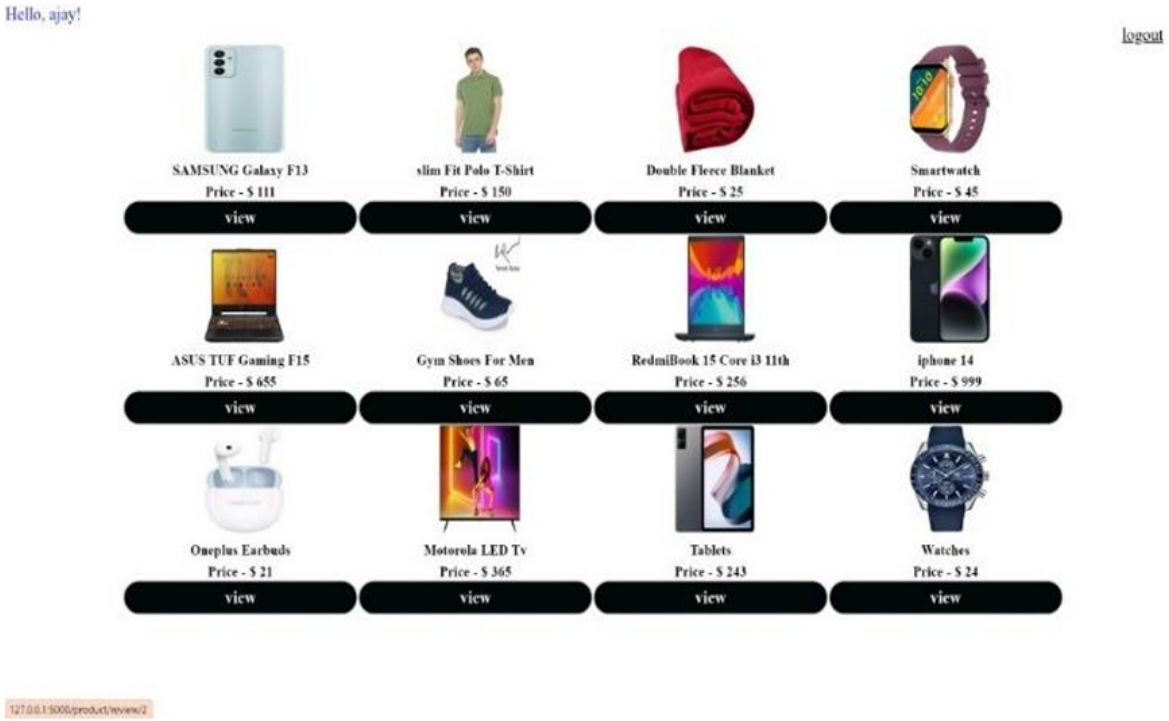

Figure 3. Website Home Page

After logging in user is directed to the above displayed website home page(Figure 3) where he can see the list of products available.

Figure 4. Products Page

The above screen (Figure 4) is accessible to the admin only and it gives the reviews given for different products in product page.

CONCLUSION

In this study, we have emphasized the importance of reviews and their impact on almost every aspect of web-based data. Reviews have a significant influence on people's decisions, making it crucial to detect fake reviews. This study proposes a machine learning-based method for detecting fake reviews, taking into account both the characteristics of the reviews and the reviewer's behavior. The proposed method is evaluated using a dataset collected from the internet, utilizing a range of classifiers. In the developed method, bi-gram and tri-gram language models are employed and compared. The results indicate that the Support Vector Machine (SVM) classifier outperforms the other classifiers in identifying fake reviews. The study also suggests that considering the behavioral characteristics of reviewers can enhance the performance of the proposed method. Although the current work considers some of the reviewer's behavioral characteristics, future research could integrate more behavioral elements, such as the frequency of a reviewer's reviews, the time taken to complete reviews, and the number of positive or negative evaluations submitted. Incorporating more behavioral variables in the strategy for detecting fake reviews is expected to improve its performance.

In summary, this study proposes a machine learning-based approach for detecting fake reviews, which takes into account both the characteristics of the reviews and the reviewer's behavior. The findings suggest that the SVM classifier is effective in identifying fake reviews, and the inclusion of additional behavioral variables can further enhance the performance of the proposed method.

FUTURE SCOPE

The Present Scope is to detect the review is a Computer Generated or an Original Review given by reviewer and display it. This project can be a base foundation and it can be incorporated with any website which wants to detect the type of reviews in it. It is scalable and suitable for advancement and development. It can also be a reference study model for upcoming developers who are keen to work in this area.

REFERENCES

[1]. Lahby, Mohamed, et al. "Online fake news detection using machine learning techniques: A systematic mapping study." Combating Fake News with Computational Intelligence Techniques (2022): 3-37.

[2]. Alsubari, S. Nagi, et al. "Data analytics for the identification of fake reviews using supervised learning." CMC-Computers, Materials & Continua 70.2 (2022): 3189-3204.

[3]. Rodrigues, Anisha P., et al. "Real-time twitter spam detection and sentiment analysis using machine learning and deep learning techniques." Computational Intelligence and Neuroscience 2022 (2022).

[4]. Tufail, Hina, et al. "The Effect of Fake Reviews on e-Commerce During and After Covid-19 Pandemic: SKL-Based Fake Reviews Detection." IEEE Access 10 (2022): 25555-25564.

[5]. Wang, Ning, et al. "A fake review identification framework considering the suspicion degree of reviews with time burst characteristics." Expert Systems with Applications 190 (2022): 116207.

[6]. Salminen, Joni, et al. "Creating and detecting fake reviews of online products." Journal of Retailing and Consumer Services 64 (2022): 102771.

[7]. Wang, Ning, et al. "A fake review identification framework considering the suspicion degree of reviews with time burst characteristics." Expert Systems with Applications 190 (2022): 116207.

[8]. Demilie, Wubetu Barud, and Ayodeji Olalekan Salau. "Detection of fake news and hate speech for Ethiopian languages: a systematic review of the approaches." Journal of big Data 9.1 (2022): 1-17.

[9]. Ahmed, Naeem, et al. "Machine learning techniques for spam detection in email and IoT platforms: analysis and research challenges." Security and Communication Networks 2022 (2022).

[10]. Kumar, Ajay, et al. "Fraudulent review detection model focusing on emotional expressions and explicit aspects: investigating the potential of feature engineering." Decision Support Systems 155 (2022): 113728.

Comparative Analysis of Stock Price Prediction by ANN and RF Model

Lopamudra Hota1, Prasant Kumar Dash2*
1Department of Computer Science and Engineering, National Institute of Technology, Rourkela, India.
2Department of Computer Science and Engineering, C. V. Raman Global University, Bhubaneswar, India.

ABSTRACT

The elementary goal of this paper is to predict the best model for estimation of stock market. Machine Learning is a blooming field in computer science that has contributed to many predictions and analysis-based algorithm in Financial and economical field. Some of the algorithms used for predictions are Random Forest (RF), Support vector machine (SVM), Long-Short Term Memory (LSTM), Artificial Neural Networks (ANN). Random Forest is an ensemble supervised learning algorithm for classification problems with high accuracy factor. ANN has matured to a great extend over the past years. With the advent of high-performance computing ANN has assumed tremendous significance and huge application potentials in recent years. The innovation of ANN technology mimics the large interconnections and networking that exists between the nerve cells to process complex task. The paper has presented ANN and RF model for stock price estimation based on historical data and computed the future price, with comparative result analysis of their performance.
Further, a candlestick model is designed of the stock to show the variation in price of stock over a stipulated period of time.

Keywords : ANN, Candle-stick, CNN, Deep Learning, Random Forest, RNN, Support Vector Machine.

INTRODUCTION

The unpredictable dynamic nature of stock market has given rise to challenges for future analysis of the stock based on present value. The Efficient-Market Hypothesis (EMH) states that prices of asset reflect all the information available. Implicating that it is not possible to "beat the market" steadily based on risk adjustment, as stocks prices in market are sensitive to updated information in regular basis.

Furthermore, the fluctuation in stock prices rely on economic, political, psychology and expectations of investors, price of commodities and many other factors. The arbitrariness and turbulences in market on every day basis led off challenges in future prediction of stocks. The accuracy in stock price and asset's future trend prediction aids in minimizing risk and maximizing profit. Various parameters are considered for stock price prediction based on statistical data computation and collection [1]. There are two aspect of predictions of asset based on past price and trends. The future prediction of stock gives a clear scenario of estimating economic status of the country. The growth of economy of a country is proportional to stock capitalization [2]. The dynamic, varied parameters, and non-linear nature of stock values generally weaken the performance analysis of statistical models for accurate prediction of future value [3].

One of the definitions of Machine Learning that states "Algorithms that parse data, learn from that data, and then apply what they've learned to make informed decisions". Machine learning models basically train machines to learn the training dataset and test the dataset trained for future prediction. Machine

learning algorithms are significantly used for performance analysis based on case studies. It identifies the data pattern and validate information to compute it as reference for future prediction and analysis.
Mostly stock market prediction is done by implementation of stochastic or random walk methodologies by estimation of future price based on successive previous price inspection and examination. Some of the common machine learning algorithms for prediction are Random Forest, Boosting and Bagging. The modernized trend in ML algorithms incorporated the Deep Learning (DL) concepts, based on non-linear topology is used in financial series of services for future trends and estimations [4].
Random Forest (RF) is one of the best models for prediction on tabular data. It takes the capabilities of multiple Decision Trees (DT) model (one of the base methods for prediction) in which the machine learns the tree having maximum influence and sets the weight values on feature points. A decision is made by branch traversal depending on various parameters. There is a problem of overfitting that arises in DT which increases the depth of the tree making it a complex structure. To overcome this problem Radom Forest is designed that is an ensemble supervised ML algorithm. The DT is created based on random features of datasets, and finally the RF is created based on the outcome of DT that is fetched maximum time.

DL is nothing but a subset of ML with variance in capabilities. DL uses a layered logical structure for estimation designed in accordance to human nervous system called Artificial Neural Network (ANN). As per the scenario of stock market, the future estimation of market is not only based on present or latest datasets but also have to deal with historical dataset for prediction accuracy of trained model. Artificial Neural Network (ANN), Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM) are some of efficient DL models for economic and financial estimation and statistical analysis [5]. Ensemble Learning model are reliable models used in today's scenario for performance and statistical prediction with minimization of overfitting issues (causes failure in accuracy of future prediction).

The tree-based models such as Decision Tree, Random Forest, Bagging, AdaBoost, XGBoost, Gradient Boosting are commonly used supervised learning ensemble estimator models for stock market prediction. ANN provides a flexible framework for computation including a high range of time series with good accuracy in future prediction. In view of this advantage of ANN we have modelled our proposal for stock market prediction based on ANN. Provided ANN has a less processing power consumption and efficient computational resource utilization, whereas Random Forest is best model for tabular data unlike ANN that can be implemented on audio, video and images as well.

REVIEW OF LITERATURE

Isaac et. Al in [6] have proposed a novel "homogeneous" ensemble classifier implemented with Genetic Algorithm (GA) for SVM parameter optimization and feature extraction for prediction stock price of Ghana stock exchange (GSE) for ten days. Authors use Decision Tree, Random Forest and Neural Network model for prediction analysis and compared the accuracy of these models. It was proved that the proposal of thew author based on Genetic Algorithm concept provides better accuracy of nearly 93.7 percent compared to other algorithms. Similarly, in [7] authors have compared the performance analysis of Neural Network models based on multiple linear regression (MLR), Elman, Jordan, Radial Basis Function (RBF), and Multilayer Perceptron (MLP). This was tested on six most traded stocks of Brazilian Stock Exchange based on RSME, number of inputs and hidden layer. As per the result MLR being the simplest model of all showed good accuracy in prediction with less computational overhead.

The researchers have proposed many models and algorithm with Machine Learning capabilities for prediction of stock prices and financial domain in past as well as today. ML is found to be one of the most powerful algorithms for information validation and future data pattern prediction. The ensemble methods in ML outperforms many of the traditional methods in time series prediction [8]. Two of the popular algorithms for prediction problem are Boosting and Bagging.

Financial data analysis is one of the most blooming topics. Computing techniques such as Neural Networks are designed for analysis of buy and sell on daily basis with less time consumption and accuracy. Many proposals have been proposed for substantial predictions [9-12].

Improved Bacterial Chemotaxis Optimization (IBCO) with ANN proposed by Zhang et.al [13] satisfies the prediction of stock price for short time of one day as well as long-time of fifteen days. Asadi et. Al [14] have proposed a Feed Forward Network along with Genetic Algorithm and Levenberg–Marquardt (LM) for learning model, basically their model is reliable dealing with the fluctuations of stock market with the capability of pre-processing data and selection of input variables. Jigar et al. [15] laid down efforts by implementing ML techniques incorporating hybrid combination of models for specifically Indian stock market index analysis and prediction. Similarly, Linear Regression model has also played an important role stock market prediction and analysing market behaviour [16].

Emioma et. al [17] proposed a model based on least square regression for prediction of future stock price handling the random changes. Some of the researchers have also used Support Vector Machine (SVM) which is a classifier that discriminate the datasets for stock market prediction [18]. Prediction of time-series has been done by methodologies based on Convolutional Neural Networks (CNN) [19], Similarly, in [20] stock market forecasting is done by using Artificial Neural Networks and Wavelet Transformations. In [21] authors have incorporated an Artificial Fish Swarm Optimization technique in neural network to achieve a more accurate model for stock prediction, taking the dataset of Shanghai Stock Exchange. Random Forest with LSTM has also been implemented to get one of the best prediction results for stock market in [22]. In [23], authors have proposed an LSTM model for stock market prediction with computation of Linear Regression.

They have used K-NN classifier for classification of dataset, computed the moving average of the stock TITAN and NIFTY50. They have used K-Means clustering concept to provide a data-frame of the stock. Bollinger bands tool is used to measure volatility of stock to help brokers and investors to predict their value of stock in future.

PROBLEM STATEMENT

The prediction of stock market is an essential factor to determine future value and movement of stock in financial sector. The more the accuracy in the share price prediction the more it will lead to profit making for investors as they get to do more accurate market analysis and study stock price pattern. This is basically done by utilizing and analysis of datasets by implementing on various models based on ML, DL, and other prediction and detection techniques. The fluctuations in the market mostly depends on the sell, buy and opinions of share-holders. The fluctuations are very rapid as the market and financial sector is fully dependent on political and social news and occasions which plays the trick for changes in stock price. As recently there has been swing of prices on day-to-day basis due to outburst COVID-19, vaccines, closing and opening of financial, economic and other sectors which drastically impact the country's economy has also a direct impact on stock market.

With a brief analysis of stock market, we have tried to predict the future stock market by implementing ANN and RF mechanism with pre-defined python libraries taking the datasets of 5-6 years from Yahoo Finance of TELA stock. The proposal has demonstrated the use of ANN and RF in prediction along with

the implementation details and outcome that we got after the implementation in the remaining section of the paper. Finally, the accuracy of algorithm for predicting the stock price is compared.

METHODOLOGY

ANN Model

Our proposal is based on designing multilayered ANN model for an accuracy efficient stock market prediction scheme. ANNs are basically single or multi-layer network structure in a fully connected fashion with input, output and hidden layers nodes. In ANN model the input in each layer nodes depends on the output of previous layer nodes i.e N depends on N-1.

As we say Artificial Neural Network, it strikes the mind that what is natural neural network? And that is the human brain which is a highly complex, nonlinear and does parallel computations with organised structural constituents "Neuron". The neuron is interconnected in a complex way between each other and one to another that gives visualization of network structure that is complex, non-linear and massively parallel. The working of ANN follows a node with weighted sum of inputs, then summed to a bias value, and passed through an activation function (non-linear function). The result that is output of this node becomes input of node in the next layer. It continues in a chain fashion and then the final output is retrieved after the result travelled the intermediate hidden layers. The increase in number of hidden layers tends to deepen the network structure [5]. Associated weights and biases train the network model.

Some of the capabilities of ANN are; exploitation of non-linearity to deal more efficiently with real-world problem which is distributed in nature, Input/Output Mapping (feed the input and wait for the desired output). For example, there may be a difference between the actual output and desired output, in that scenario we can take some free parameters (weights or input strength in ANN and Synaptic connection in biological term) to minimize the margin between the actual and desired output by finding a closest value. ANN has a facility of learning in which the system can learn about the desired output to get the output closest to the desired output value, this feature makes it different from traditional computational unit, that is called the Learning ability with adaptability to certain specific characteristics due to changes in the environment. Ability for fault tolerance is another most important capability of ANN, fault being directly proportional to performance degradation.

Perceptron (Computer Neuron)

A perceptron algorithm is based on binary classifiers, that takes an input signal and provides an output signal after processing or computation, basically a single-layered neural network. The input layer resembles the dendrites of the neuron and output signal the axon. The input signals are assigned weights which are multiplied by input values, and the weighted sum of all inputs are stored in the neuron. The computation of the weights is done by gradient decent and back-propagation algorithms, these adjust the free parameters to minimize the loss/cost function. Back-propagation model are generally used in training of Feed-Forward Neural networks.

The ANN consists of hidden layers in multi-layer perceptron scenario. For the hidden or output nodes, nodes take the weighted sum of inputs, add to a bias, and then passes it through an activation function non-linear function. The result fetched at this node becomes input for another node in next layer. The process is iterated for all the nodes for determining the final output. And the network is trained by learning process of associated weights and biases of nodes.

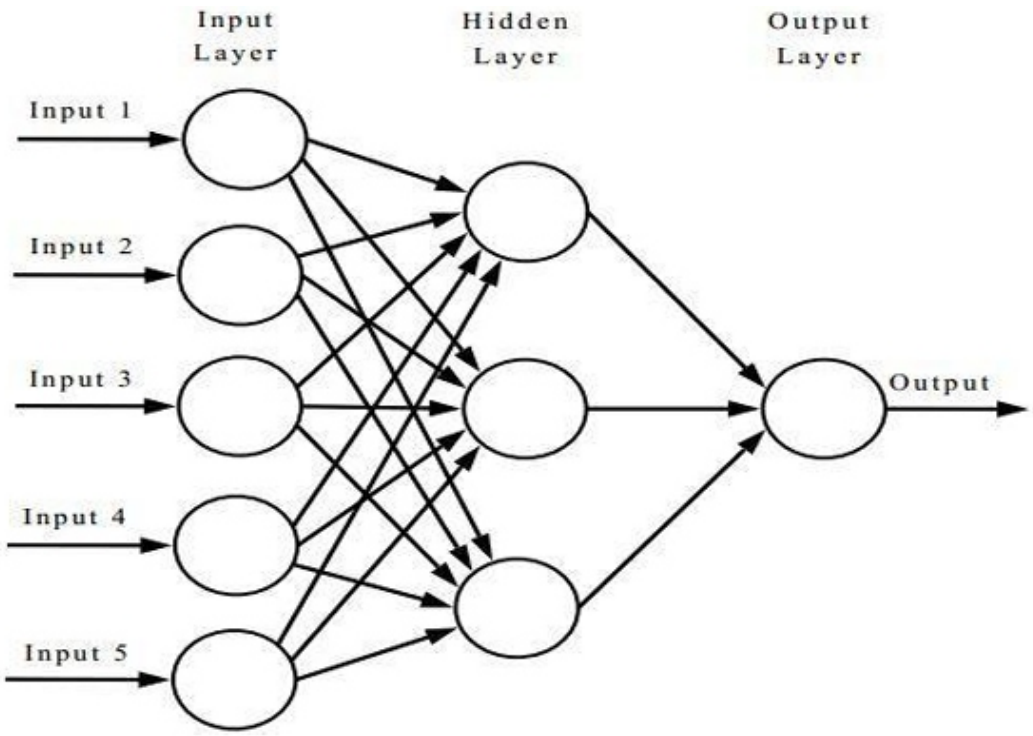

Fig 1: Schematic Structure of ANN

The equation below [6] show the relation between weights, biases and activation function for computation of the learning model.

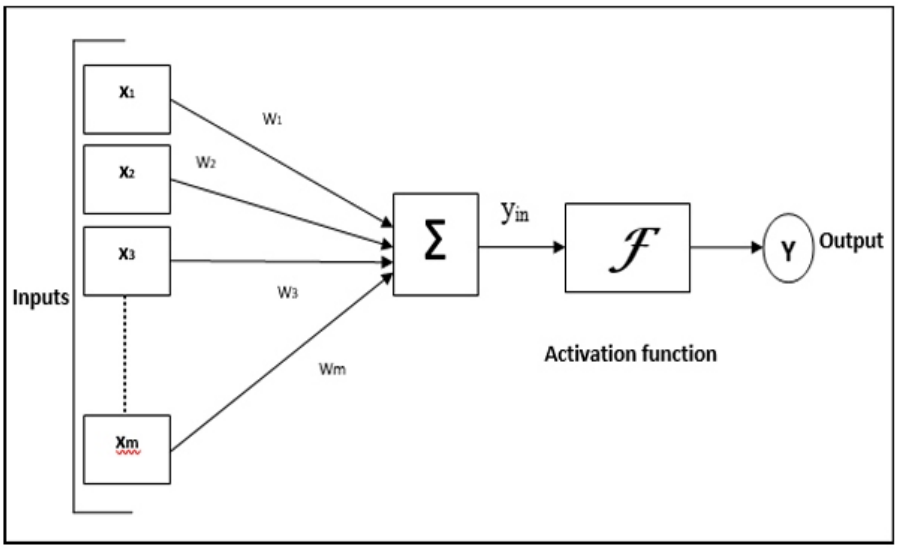

Fig 2: Working of ANN Model

It basically works same as the principle of human brain, take input values and iterate it over multiple processing steps to fetch desired result, the inputs in case of human brain can be smell, seeing, hearing, touch, and taste. The feelings and emotions are the two hidden layers through which processing is carried out that make us to take decisions (output). This gives a brief understanding although we know that computation in brain is much more complex and have many more hidden layers for processing tasks.

Random Forest Model

It is basically a ML based supervised ensemble learning mechanism used for classification as well as Regression problems. The procedure merges various classifiers to solve complex tree overfitting problem thereby improvising model performance by enhancing the prediction accuracy of datasets. Rather than considering the outcome of a single DT, it combines the outcome of all DTs and takes the outcome with majority votes to predict the end result. The accuracy of the RF is directly proportional to the number of tree structures in the forest. RF has higher accuracy, less time to train as compared to other algorithms even on larger datasets.

Here, the selection of feature is done by Bootstrap or Bagging mechanism. The training sets are created from the feature set of datasets by selecting random ones and a feature may be repeated in other training sets. This random selection procedure ensures less correlation and minimizes overfitting.

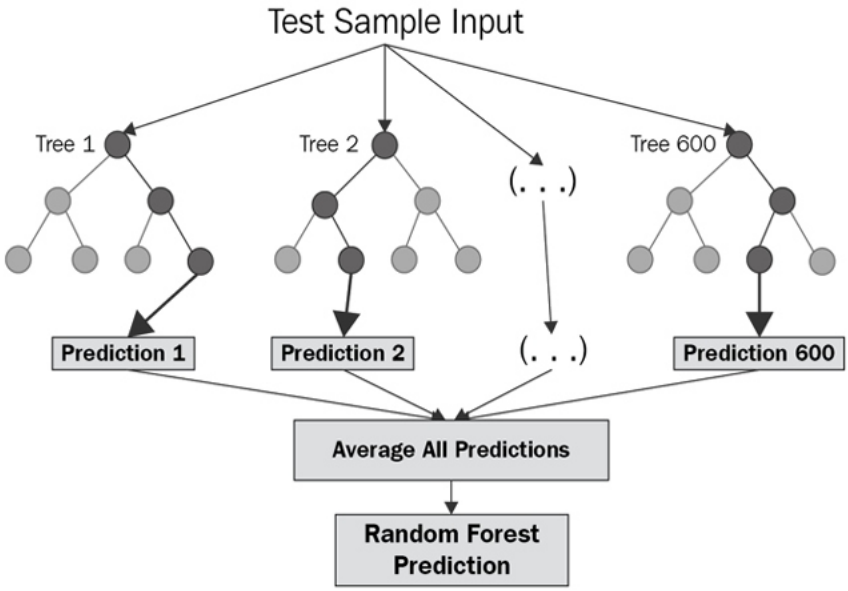

Fig 3: Schematic Structure of Random Forest

Basic Structure of Random Forest

The basic RF model contains the following working procedure to be implemented on the dataset.

1. Selection of K datapoints from training dataset.
2. Decision tree construction with the selected datapoints
3. Select N Decision Trees
4. Again process 1 and 2
5. For computation of new data-points, evaluate the predictions of DTs and assign datapoints to the ones that has maximum votes.

PROPOSED WORK

Here, we have used the discussed ANN model as well RF model for prediction of price in stock market. In ANN, as per the diagram structure there are the hidden layer which depicts open/close price and their difference, volume. The model is trained based on weight applied to an activation function for getting a specified output. The final output is calculated by sum of output compute for each neuron.

Initially the ANN train itself for the given data set specifically the Open, High, Low, Close and Volume

(OHLCV) consisting of variables; timestamp (epoch time), open price, close price, high price, low price and volume of stock (quantity of assets sold or purchased). The flat file data field consists of all historical data in .csv format. We have taken the input values as the open, close price to train the model and then compute the actual output Y, which is somewhat different from the predicted output Y'.

The weights of the variables are modified for each neuron by cost function minimization, that defines the cost of prediction making, that is gap between the actual output value and predicted output. Basically, cost function measures "how acceptable" a neural network is with respect to the training sample given and the output expected. We have computed the cost function as the sum squared deviation between that of computed actual output and predicted output values. Initially the cost function is computed for the given datasets with a specified set of weights for each neuron. Neural networks learn according to the weights associated with the neuron which is updated after every forward passage of data through the neural network structure. The weights are adjusted to aid in reconciliation of the differences between actual value computed and predicted output for consecutive forward propagation. Question arises, how can we adjust these random weights assignment? As we have already discussed about the differences in actual and predicted outcomes, the error factor becomes an important consideration for computation of weights. At each neuron errors are computed and the reverted back to the neurons through the network by Back-Propagation for update of weight facilitated by forward pass. The procedure is repeated till we achieve a minimized cost factor.

In RF model, we fetch the dataset then create the input variables based on Open-Close and High-Low values and output variables by setting it to 1 if next day closing price is greater than today's and -1 otherwise, done randomly. The decision is made by dividing data into sets that are heterogeneous with others and homogeneous among them.

This division is based on some criteria like Mean Squared Error (MSE) termed as Information Gain that predicts how our model accuracy improves as we split further.

The parameters taken for RF implementation are estimators, depth value of DT, bootstrap, and number of samples. The estimator signifies the number of DT incorporated. Bootstrap pick the random values from samples for maximized accuracy and visualization of result by building multiple DT models. RMSE is computed to estimate the best DT computed from the dataset. For each DT, fetch a subset of training dataset randomly and fit the DT. Repeat for all DTs created. As the subset are randomly chosen computer the error for each model and then by taking average of these model the information is merged to get the predicted or computed dataset.

Design of Trading Model:

We have implemented our model by the use of Neural Network and Random Forest methodologies in python for stock market future estimations and accurate prediction for trader's welfare. Python, use powerful libraries for building robust, efficient and reliable Trading Model. We have used predefined python libraries like numpy and pandas for dataset computations. Talib for computation of Relative Strength Index (RSI) (value between 0 and 100) and William %R Oscillator (value between 0 and -100) in ANN. Williams %R Oscillator computation has a significance in measuring closing price for a specified time period within the trading range. RSI computes the consistency of variation in prices over time, so high value of RSI signifies frequent increase in price than the declined rate over a time duration. RandomForestRegressor and bootstrap with quantrautil and RandomForestClassifier module for Random forest computation.

The data set has been imported from Yahoo Finance, and we have implemented our model for TESLA datasets to compute the stock return and planning return of a stock invested by the investor. These are computed by initially computing the return for the next day, for 'true' predicted value long leaps are

taken whereas for ‘false’ value shorter leaps are taken. The rate of return is computed, if long leaps are present at the end of a day, squared at the end of the next day. After computation of return the cumulative return values are calculated, based on this value a graph is plotted to demonstrate how our planning return performs against the stock return. We have also implemented a candlestick model that estimate the possible price movement based on past values for a specified time period of one year form MAR 2020 to MAR 2021.

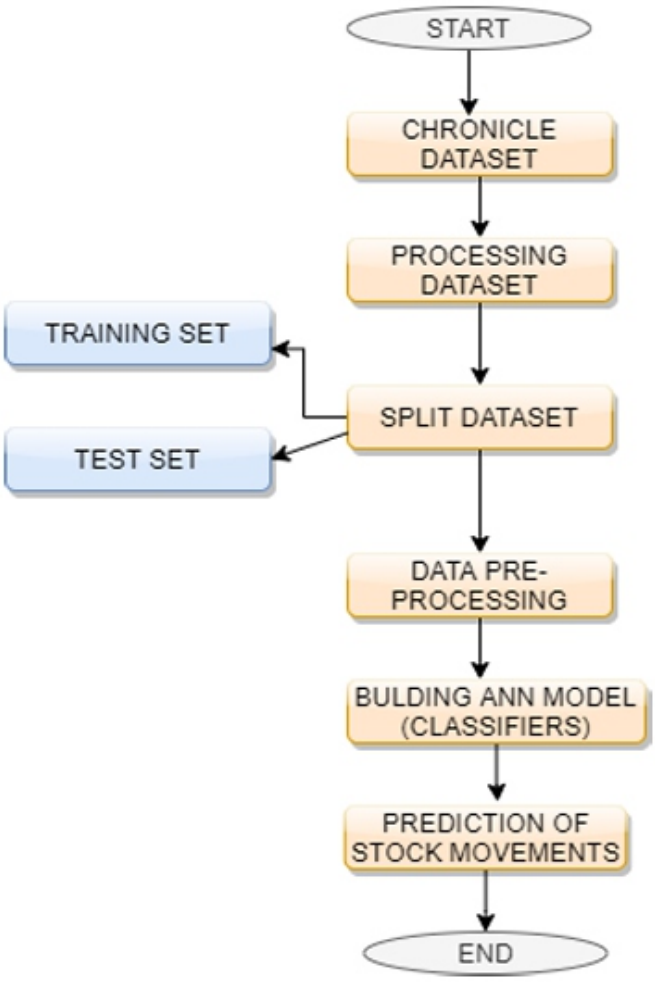

Fig 4: Flow Chart for ANN

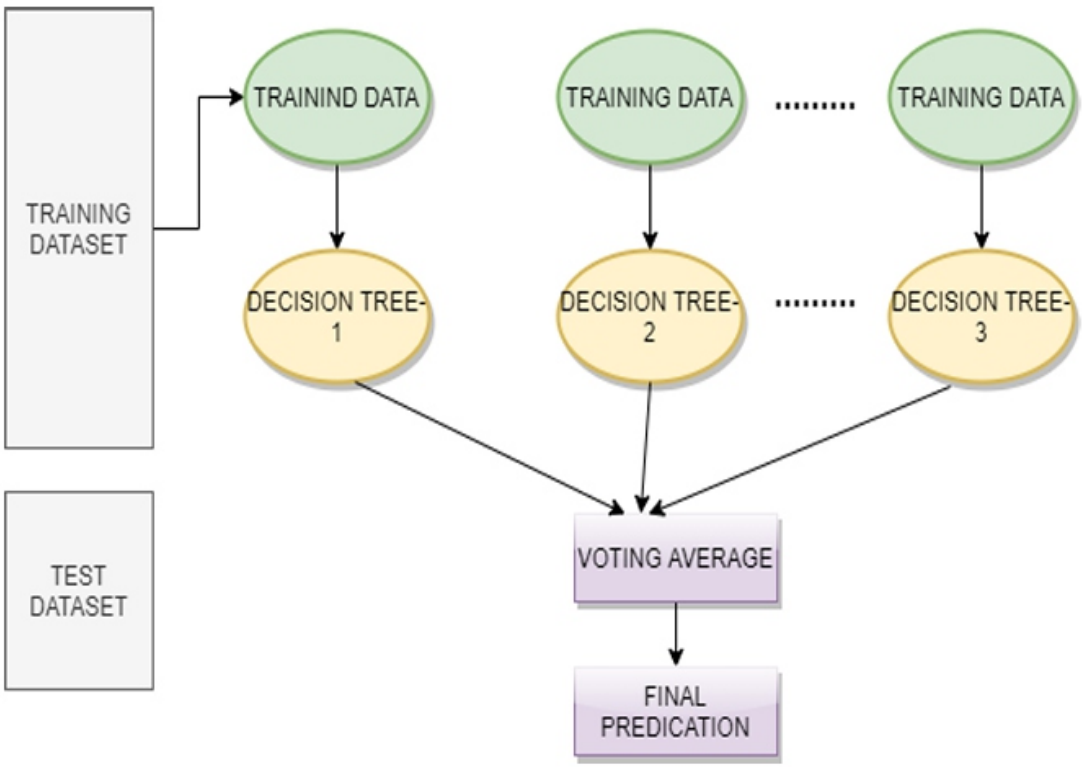

Fig 5: Flow Chart for RF

EXPERIMENTAL RESULTS

The data has been extracted from Yahoo Finance of TSLA stock with seven attributes stating the high-low, close-open values of stock. The dataset was implemented for ANN as well as RF for prediction of stock price and analysis of accuracy based on actual and predicted value. The adjusted close price of the specified duration is computed and graph is plotted. A candlestick model of the stock is also implemented to record the fluctuation in the stock price in a specified duration (MAR 2020 to MAR 2021).

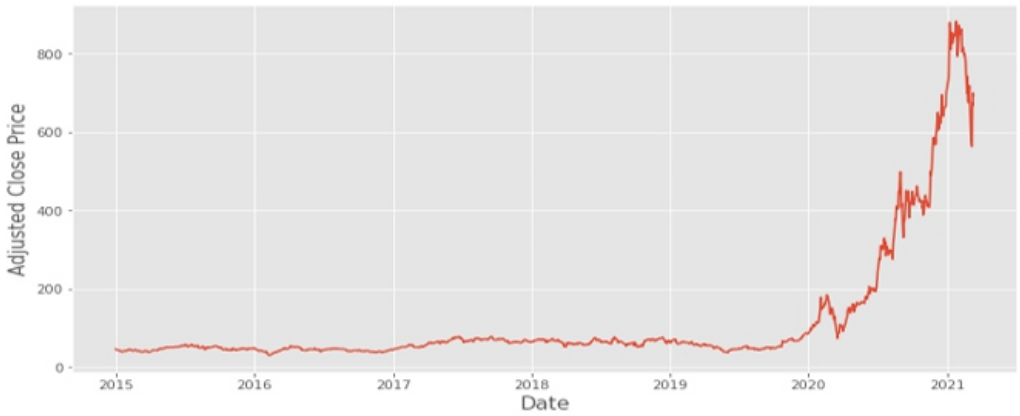

Fig 6: Graph for Adjusted Close Price (RF model)

Fig 7: Candle-Stick Chart

In the RF model, the confusion matrix was built to examine various possible outcome of prediction to reach at an accurate prediction. It predicts the correctness and in-correctness of the classifier's prediction giving the accuracy of the model. Graph is designed for real and predicted price over time vs price.
Along with the same dataset, the ANN is modelled and stock prediction is made with computation of buy/sell profit, cumulative profit and predict the future price after 15days with accuracy of 82.8% and MSE of 5.89. Graph is designed for actual and predicted over year vs price.

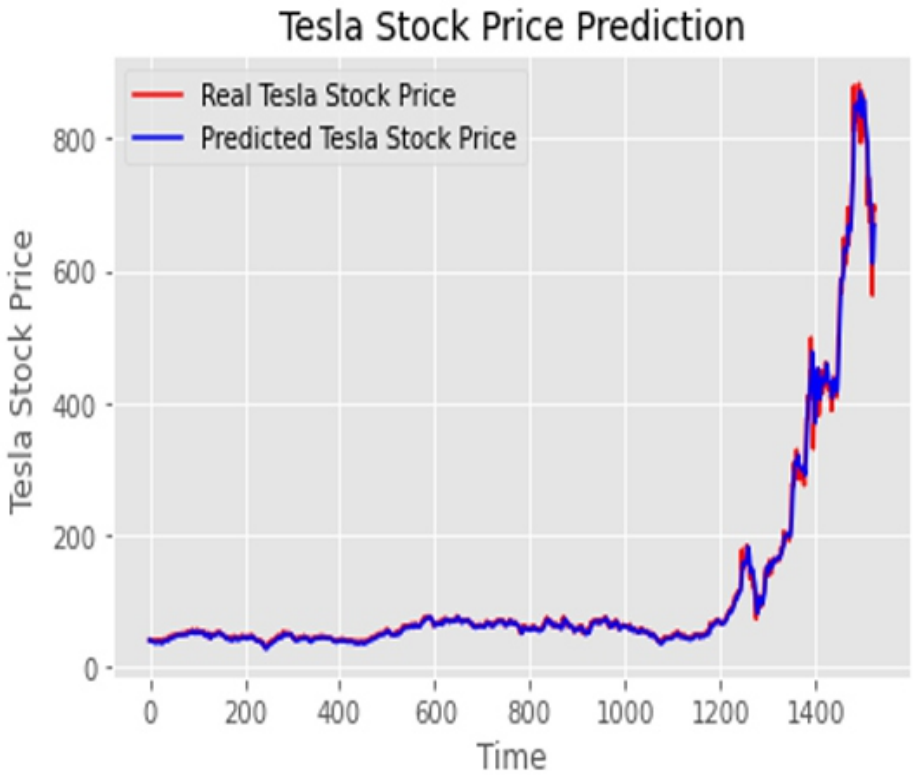

Fig 8: Graph Stock Price vs Time (RF)

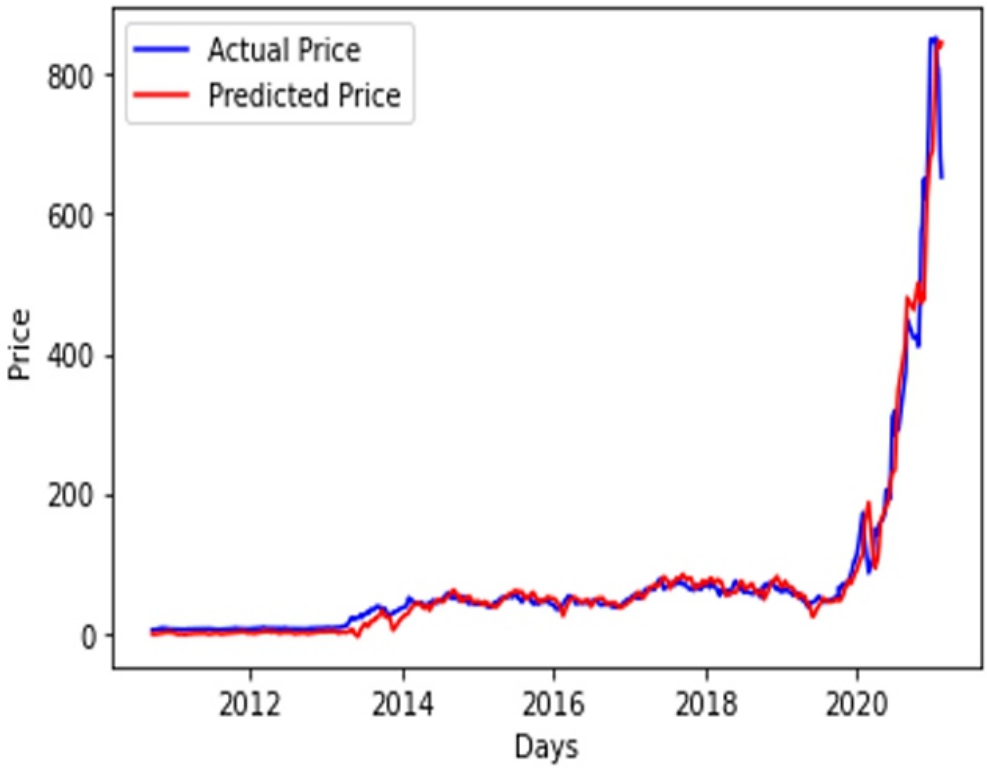

Fig 9: Graph Price vs Days (ANN)

By measure of various parameter and accuracy computation it is found that although Random Forest is a better model for financial and stock prediction based on historical dataset still, the advanced Artificial Neural Network model shows better result with RSME between 4 to 7. Although every ML algorithm has its own way and perform as they are designed; still Neural Network models are marginally better in performance that traditional ML models whether it is textual data for prediction or audio/visual data for recognition.

Table 1: Comparison of Algorithm

Accuracy of Algorithm	
Random Forest (RF)	79.6
Artificial Neural Network (ANN)	82.8

CONCLUSION AND FUTURE SCOPE

In the dilemma of choosing the best model for stock market or any financial prediction, the decision should be made based on data and parameters for ease of computation. Although Neural Network performs better in all type of data values it can be specifically used in audio, video and pictorial dataset. Contrast to this model like Random Forest, Decision Tree, Linear Regression and Support Vector

Machine can do a good job with tabular data due to their simplicity in computation and implementation thereby minimizing the overhead to work with complex structure of Neural Network. Still there is always an opportunity to switch to neural network to achieving better performance and accurate prediction.

For future work, we would like to implement our model on larger datasets and also implement the datasets using Recurrent Neural Network (RNN) and Convolution Neural Network (CNN) with LSTM and compare with result of ANN. The use of Deep Reinforcement Learning (DRL) with Q-Learning to get an optimized result and able to predict financial and stock market with better accuracy can be done in a simpler way with lesser historical data. The DRL models an intelligent system to handle fluctuations and test updated actions and approaches; keeping a check on failure and success rates in a continuous basis.

REFERENCES

[1] J. Lehoczky, M. Schervish, Overview and History of Statistics for Equity Markets. Annu. Rev. Stat. Its Appl. 2018, 5, 265288.

[2] A. Aali-Bujari, F. Venegas-Martínez, G. Pérez-Lechuga, Impact of the stock market capitalization and the banking spread in growth and development in Latin American: A panel data estimation with System GMM. Contaduría y Administración 2017, 62, 1427–1441.

[3] M.P. Naeini, H. Taremian, H.B. Hashemi, Stock market value prediction using neural networks. In Proceedings of the 2010 international conference on computer information systems and industrial management applications (CISIM), Krakow, Poland, 8–10 October 2010; pp. 132–136.

[4] R.C. Cavalcante, R.C. Brasileiro, V.L. Souza, J.P. Nobrega, A.L. Oliveira, Computational intelligence and financial markets: A survey and future directions. Expert Syst. Appl. 2016, 55, 194–211.

[5] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, Deep learning for Stock Market Prediction. arXiv 2020, arXiv:2004.01497.

[6] Isaac Kofi Nti, Adebayo Felix Adekoya, and Benjamin Asubam Weyori, Efficient Stock-Market Prediction Using Ensemble Support Vector Machine, Open Computer Science, 2020; 10:153–163, https://doi.org/10.1515/comp-2020-0199.

[7] S. Teixeira, Z. Pauli, M. Kleina, and W. H. Bonat, Comparing Artifcial Neural Network Architectures for Brazilian Stock Market Prediction, Annals of Data Science, 2020, 7(4):613628 https://doi.org/10.1007/s40745-020-00305-w.

[8] S. Amari, The Handbook of Brain Theory and Neural Networks; MIT press: Cambridge, MA, USA, 2003.

[9] M. Ballings, et al., Evaluating multiple classifiers for stock price direction prediction. Expert Systems with Applications, 2015. 42(20): p. 7046-7056.

[10] E. Hadavandi, H. Shavandi, A. Ghanbari, Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting. Knowledge-Based Systems, 2010. 23(8): p. 800-808. 20.

[11] Y.S. Lee, and L.I. Tong, Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming. Knowledge-Based Systems, 2011. 24(1): p. 66-72.21.

[12] M.F. Zarandi, E. Hadavandi, and I. Turksen, A hybrid fuzzy intelligent agent-based system for stock price prediction. International Journal of Intelligent Systems, 2012. 27(11): p. 947-969.

[13] Y. Zhang and L. Wu, Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network. Expert systems with applications, 2009. 36(5): p. 8849-8854.

[14] S. Asadi, et al., Hybridization of evolutionary LevenbergMarquardt neural networks and data pre-processing for stock market prediction. Knowledge-Based Systems, 2012. 35: p. 245-258.

[15] J. Patel, et al., Predicting stock market index using fusion of machine learning techniques. Expert Systems with Applications, 2015. 42(4): p. 2162-2172.
[16] S. Abdulsalam, K.S. Adewole, and R. Jimoh, Stock trend prediction using regression analysis–a data mining approach. 2011.
[17] C. C. Emioma and S. O. Edeki, Stock price prediction using machine learning on least-squares linear regression basis, International Conference on Recent Trends in Applied Research (ICoRTAR) 2020, doi:10.1088/1742-6596/1734/1/012058.
[18] Z. Guo, H. Wang, Q. Liu, and J. Yang, A feature fusion based forecasting model for financial time series. PloS one, 9(6):e101113, 2014.
[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Image-net classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.
[20] W. Bao, J. Yue, and Y. Rao, A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PloS one, 12(7):e0180944, 2017.
[21] W. Shen, et al., Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm. Knowledge-Based Systems, 2011. 24(3): p. 378-385.
[22] S. Han, Stock Prediction with Random Forests and Long Short-term Memory, 2019, https://lib.dr.iastate.edu/creativecomponents.
[23] A. Singh , P. Gupta, and N. Thakur, An Empirical Research and Comprehensive Analysis of Stock Market Prediction using Machine Learning and Deep Learning techniques, 2021, IOP Conf. Series: Materials Science and Engineering 1022 (2021) 012098, doi:10.1088/1757-899X/1022/1/012098.

Enhancing GraphQL Authorization with Open Policy Agent (OPA)

Venkata Thota
Solution Architect/Lead

ABSTRACT

GraphQL has developed into a powerful query language for APIs, allowing for unprecedented flexibility when retrieving data. However, securing GraphQL APIs, especially when it comes to authorization, poses one of the most challenging tasks. This paper explores how Open Policy Agent (OPA) serves as a robust solution to address these challenges by providing a unified policy language for access control across diverse services, including GraphQL. In the document, GraphQL authorization is explored, emphasizing its distinct challenges compared to traditional REST APIs. Due to GraphQL's dynamic nature and the ability of clients to specify the exact data they wish to retrieve traditional access control mechanisms have difficulty providing fine-grained authorization controls. Open Policy Agent (OPA) is a general-purpose policy engine that is open-source and contains a declarative policy language known as Rego. By using this language, developers are able to articulate complex authorization logic in a concise and clear manner. A step-by-step procedure is provided for integrating OPA with GraphQL, providing guidance on defining policies in Rego, integrating OPA into the GraphQL server, and enforcing fine-grained authorizations. This document discusses how to handle complex relationships, nested queries, and the importance of auditing and monitoring authorization decisions. The benefits of implementing GraphQL authorization with OPA are highlighted, emphasizing consistency, flexibility, and scalability. The document concludes with sample Rego policies that can be used as a foundation for securing GraphQL services, catering to various authorization scenarios such as authentication, depth limitation, role-based access, and field-level restrictions.

Keywords : Open Policy Agent (OPA), Authentication, Authorization, Fine-Grained, GraphQL Security, Apollo Router

INTRODUCTION

GraphQL, a powerful query language for APIs, has gained immense popularity for its flexibility and efficiency in data fetching. However, securing GraphQL APIs can be a complex challenge, especially when it comes to authorization. Open Policy Agent (OPA) [1] offers a robust solution to address these challenges by providing a unified policy language for access control across various services, including GraphQL.

Understanding GraphQL Authorization:

Authorization in GraphQL involves determining whether a user or a client has the necessary permissions to execute a specific query or mutation. Unlike traditional REST APIs, where endpoints may correspond to specific actions, GraphQL exposes a single endpoint for all interactions, making fine-grained [2] authorization a crucial aspect of securing the API.

Challenges in GraphQL Authorization:
GraphQL's flexibility in query construction allows clients to request precisely the data they need. This presents a challenge for traditional access control mechanisms, as the authorization logic must account for the specific fields requested within a query. Additionally, handling complex relationships between types and ensuring consistent authorization across various operations further complicates the authorization process.

Open Policy Agent (OPA):
Open Policy Agent (OPA) [1] is an open-source, generalpurpose policy engine that enables fine-grained, context aware access control across diverse software stacks. OPA [1] uses a declarative policy language called Rego, which allows user to express complex authorization logic in a clear and concise manner.

Architecture:
The architecture depicted in Figure 1 provides a complete solution for Authentication and Authorization. It utilizes JSON Web Tokens (JWT) [3] to secure and manage access to resources. This design ensures a strong and scalable system, which enhances the security of applications and services. The following steps explain the architecture flow for both authentication and authorization.

Step 1: GraphQL Request with ClientID Header
• The GraphQL client sends a request to the Apollo Router, including the clientId as a Header.

Step 2: Apollo Router Processing and Coprocessor Authentication
• Apollo Router processes the incoming request.
• The Coprocessor, an extensibility of the router, operates as a sidecar.
• The Coprocessor retrieves the client secret from Vault [4] using the ClientID.
• Coprocessor initiates an authentication request to the JWT [5] token provider:

1. Pulls the client secret from Vault [4] using the ClientID.
2. Performs the authentication request using ClientID /Secret.
3. Token Provider authenticates using ClientID /Secret and generates a JWT [5] Token if successful.
4. Sends the generated JWT [5] Token back as a response to the Coprocessor.

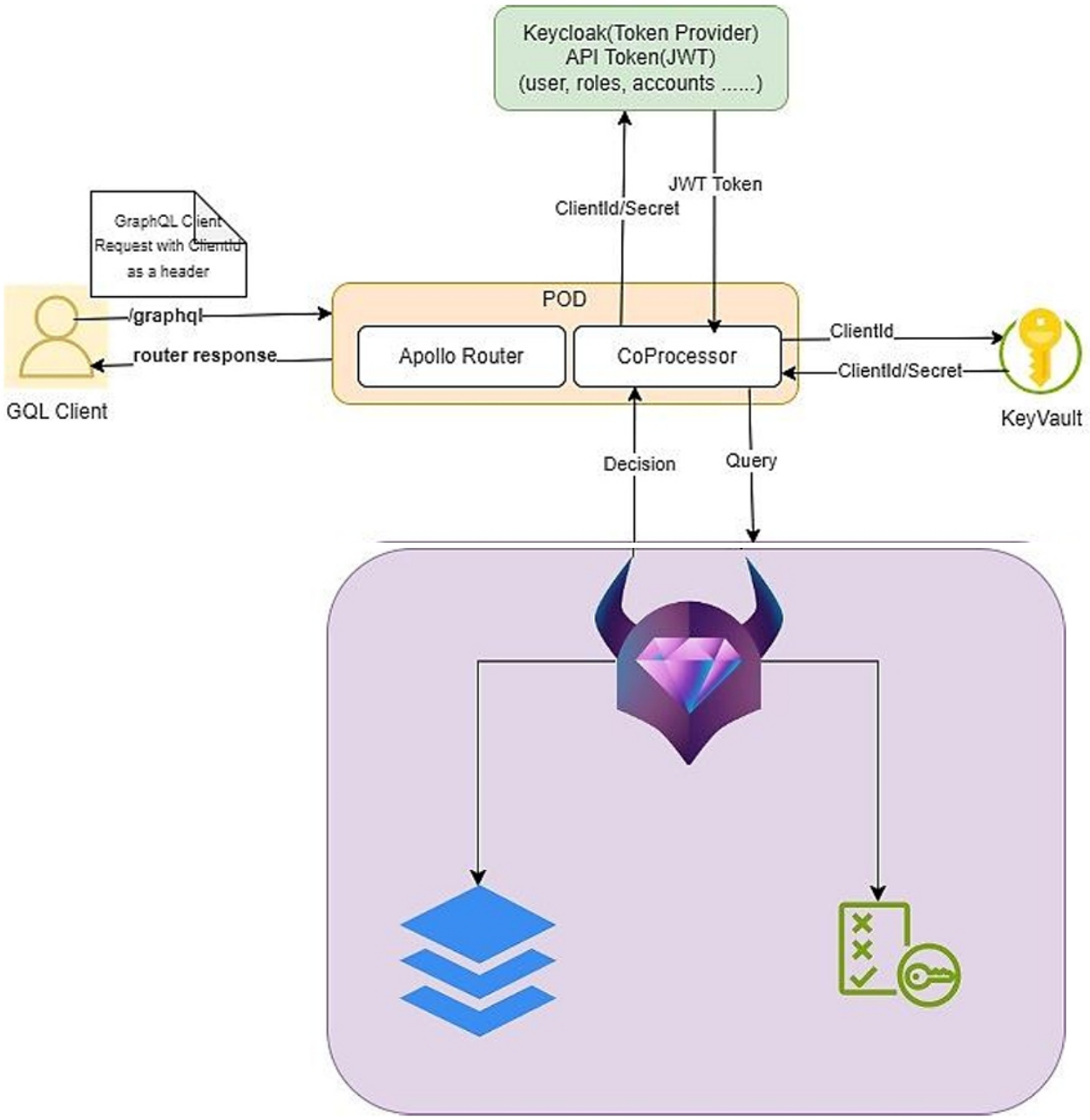

Figure 1. Architecture Diagram for Authentication and Authorization

Step 3: JWT Token Extraction and Authorization Request

- Coprocessor extracts the JWT Token from the received response.
- Prepares an authorization request to the Open Policy Agent (OPA).

Step 4: OPA Validation and Decision

- OPA [1] receives the authorization request along with the JWT [5] Token.
- OPA [1] validates the JSON request against the defined policy.
- Provides a decision (allow or deny) based on the policy evaluation.

Step 5: Response to GraphQL [6] Client

• Apollo Router receives the decision from OPA [1].

• If allowed, Apollo Router proceeds with executing the GraphQL [6] request.

• If denied, Apollo Router sends an appropriate response to the GraphQL [6] client, indicating access denial.

This architecture flow outlines the process from the GraphQL client request, through authentication using JWT [5] tokens, to authorization using OPA, and finally, the response to the GraphQL [6] client based on the policy decision.

OPA Execution Flow:

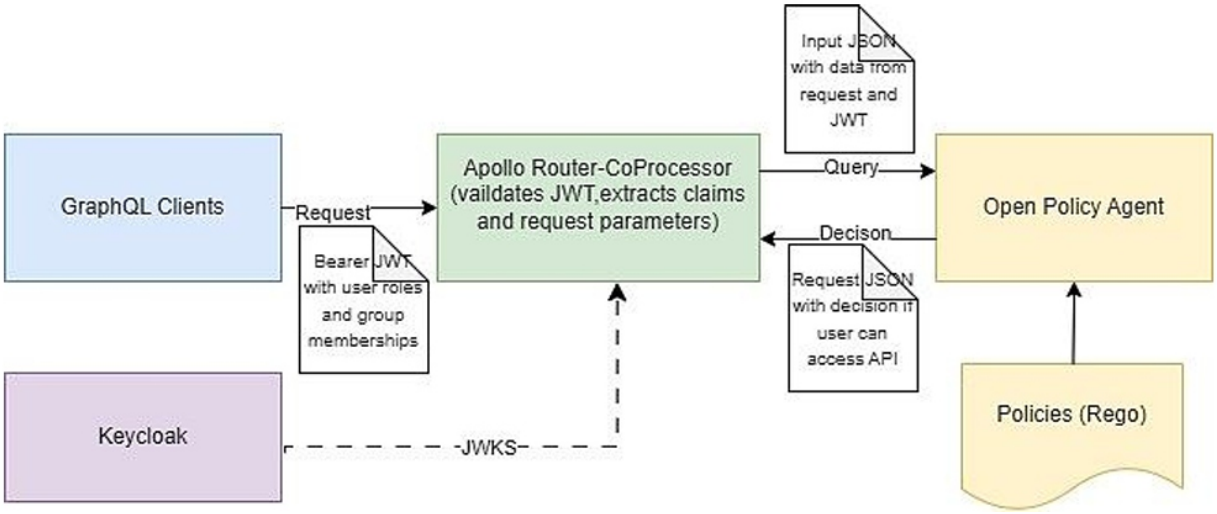

Figure 2. Integrating OPA with GraphQL Router

Figure 2 illustrates the implementation of GraphQL authorization using OPA. To achieve this, follow these key steps:

Define Policies in Rego [7]:

Write policies in Rego [7] that express the authorization logic for GraphQL API. These policies can include rules for specific queries, mutations, and fields based on user roles, attributes, or any other relevant context.

Integrate OPA [1] into the GraphQL Server:

Integrate OPA [1] into your GraphQL server to evaluate policies at runtime. This can be done by creating a middleware or a resolver that intercepts incoming requests and consults OPA [1] for the authorization decision.

Enforce Fine-grained [2] Authorization (FGA):

Leverage OPA's ability to understand the structure of GraphQL queries to enforce fine-grained [2] authorization. OPA can inspect the requested fields and relationships within a query, ensuring that users only get access to the data they are authorized to retrieve.

Handle Relationships and Nested Queries:
GraphQL's nested structure allows clients to request data at different levels of depth. OPA can handle these relationships by recursively evaluating authorization policies for each level of the query, providing a comprehensive and secure approach to nested queries.

Audit and Monitor Authorization Decisions:
OPA provides transparency into the authorization process by logging decision details. Use this information for auditing and monitoring purposes, allowing you to track and analyze access patterns, identify potential security threats, and make informed policy adjustments.

Benefits of GraphQL Authorization with OPA:
• Consistency: Ensure consistent and centralized authorization logic across your GraphQL API.
• Flexibility: Adapt authorization policies easily without modifying the underlying GraphQL server.
• Scalability: Handle complex access control requirements as your GraphQL schema evolves.

Rego [7] Policy Language:
Rego [7] (short for "Regulation") is a policy language used with Open Policy Agent (OPA) to enforce policies across cloud-native environments. Below are some sample Rego [7] policies that user can use as a starting point for securing GraphQL services. These policies assume that user has a basic understanding of Rego [7] and OPA.

Sample Policy 1: Allow only authenticated users to access certain GraphQL operations package graphql.security
default allow = false
allow { input.request.method == "POST" input.request.path == ["graphql"] input.parsedToken != }

This policy ensures that only authenticated users can perform GraphQL mutations by checking if the HTTP method is POST, the path is "/graphql", and a valid authentication token is present.

Sample Policy 2: Limit the depth of GraphQL queries to prevent abuse package graphql.security
default allow = false
allow { input.request.method == "POST" input.request.path == ["graphql"] count(input.parsedQuery) <= 10 }

This policy restricts the depth of GraphQL queries to 10 levels. Adjust the limit according to your application's needs to prevent overly complex queries.

Sample Policy 3: Allow only specific roles to execute certain GraphQL operations package graphql.security
default allow = false
allow { input.request.method == "POST" input.request.path == ["graphql"] input.parsedToken != null has_permission(input.parsedToken, "write_data") } has_permission(token, permission) { token.roles[_] == permission }

In this policy, only users with the "write_data" role are allowed to execute GraphQL mutations. Customize the role and permission checks based on your authorization requirements.

Sample Policy 4: Restrict access to specific GraphQL fields based on user roles package graphql.security
default allow = false
allow { input.request.method == "POST" input.request.path == ["graphql"] input.parsedToken != null can_access_field(input.parsedToken, input.parsedQuery) } can_access_field(token, query) { some field
field == "sensitiveField"
token.roles[_] == "admin" }

This policy restricts access to the "sensitiveField" in GraphQL queries, allowing only users with the "admin" role to access it. Extend the can_access_field rule for other sensitive fields.

Sample Policy 5: Rate limit GraphQL requests per user package graphql.security
default allow = false
allow { input.request.method == "POST" input.request.path == ["graphql"] input.parsedToken != null not rate_limited(input.parsedToken) } rate_limited(token) { count_recent_requests(token) > 10 }
count_recent_requests(token) = count { recent_request[token] = timestamp timestamp - recent_request[token] < 60 }

This policy prevents users from making more than 10 GraphQL requests per minute. Adjust the limit as needed. Remember to adapt these policies based on business specific use cases, GraphQL schema, and authentication/authorization mechanisms. Integrate these policies into OPA setup to enhance the security of GraphQL services.

CONCLUSION

To conclude, Open Policy Agent (OPA) combined with GraphQL authorization offers a flexible and robust solution to securing GraphQL APIs. As highlighted in this paper, GraphQL's dynamic nature and the ability of clients to precisely define their data retrieval requirements present distinct challenges in comparison to traditional REST APIs, making Fine-grained [2] Authorization a critical aspect.
The comprehensive architecture illustrated in Figure 1, along with the detailed step-by-step integration process, demonstrates how OPA can be seamlessly incorporated into a GraphQL server for enhanced access control. GraphQL queries require developers to articulate complex authorization logic concisely using OPA's declarative policy language, Rego [7].

As outlined above, the benefits of implementing GraphQL authorization with OPA, such as consistency, flexibility, and scalability, emphasize the advantages of this approach in ensuring a secure and reliable API. These sample Rego [7] policies cover various authorization scenarios, including authentication, depth limitation, role-based access, and fieldlevel restrictions for securing GraphQL services. By enforcing Fine-grained [2] Authorization through OPA, organizations can ensure that users receive access only to the data they are authorized to access, even in the face of GraphQL's dynamic query construction. In addition, OPA's auditing and monitoring capabilities make the authorization process transparent, enabling organizations to monitor access patterns, identify potential security threats, and adjust policies accordingly.

The integration of OPA with GraphQL authorization is in line with industry standards and the evolving landscape of API security. This provides a solution that is both powerful and adaptable to the dynamic

nature of GraphQL. With more organizations adopting GraphQL for its efficient data fetching, OPA is an important ally in enhancing the security and reliability of GraphQL APIs.

ACKNOWLEDGEMENTS

We extend our sincere gratitude to all those who contributed to the development and completion of this article on "Securing GraphQL APIs with Open Policy Agent (OPA): A Fine-Grained Authorization Approach." Special thanks to the authors and researchers who dedicated their time and expertise to thoroughly investigate and articulate the challenges and solutions in GraphQL authorization, with a focus on integrating Open Policy Agent. The collaborative effort involved in outlining the architecture, detailing the step-by-step integration process, and providing sample Rego policies has been instrumental in delivering a comprehensive resource for developers, security professionals, and organizations navigating the complexities of GraphQL security. This acknowledgment extends to the broader community that engages in the discourse around API security and GraphQL best practices. The commitment to knowledgesharing and advancing secure development practices is pivotal, and we appreciate the collective dedication that makes such contributions possible.

REFERENCES

[1] OPA Documentation: Open Policy Agent (OPA) documentation. Available at: https://www.openpolicyagent.org/docs/latest/

[2] Fine-Grained Authorization in GraphQL: Article on fine-grained authorization in GraphQL. Available at: https://blog.apollographql.com/fine-grained-authorization-ingraphql-bfd73c5153b

[3] JSON Web Tokens (JWT) Overview: JWT overview. Available at: https://jwt.io/introduction/

[4] Vault Documentation: HashiCorp Vault documentation. Available at: https://www.vaultproject.io/docs/

[5] JWT Documentation: JSON Web Tokens (JWT) documentation. Available at: https://jwt.io/introduction/

[6] GraphQL Documentation: GraphQL official documentation. https://graphql.org/

[7] Rego Language Documentation: Rego language documentation. Available Available at: at: https://www.openpolicyagent.org/docs/latest/policy-language/

Instructions for Authors

Essentials for Publishing in this Journal

1 Submitted articles should not have been previously published or be currently under consideration for publication elsewhere.

2 Conference papers may only be submitted if the paper has been completely re-written (taken to mean more than 50%) and the author has cleared any necessary permission with the copyright owner if it has been previously copyrighted.

3 All our articles are refereed through a double-blind process.

4 All authors must declare they have read and agreed to the content of the submitted article and must sign a declaration correspond to the originality of the article.

Submission Process

All articles for this journal must be submitted using our online submissions system. http://enrichedpub.com/ . Please use the Submit Your Article link in the Author Service area.

Manuscript Guidelines

The instructions to authors about the article preparation for publication in the Manuscripts are submitted online, through the e-Ur (Electronic editing) system, developed by **Enriched Publications Pvt. Ltd**. The article should contain the abstract with keywords, introduction, body, conclusion, references and the summary in English language (without heading and subheading enumeration). The article length should not exceed 16 pages of A4 paper format.

Title

The title should be informative. It is in both Journal's and author's best interest to use terms suitable. For indexing and word search. If there are no such terms in the title, the author is strongly advised to add a subtitle. The title should be given in English as well. The titles precede the abstract and the summary in an appropriate language.

Letterhead Title

The letterhead title is given at a top of each page for easier identification of article copies in an Electronic form in particular. It contains the author's surname and first name initial .article title, journal title and collation (year, volume, and issue, first and last page). The journal and article titles can be given in a shortened form.

Author's Name

Full name(s) of author(s) should be used. It is advisable to give the middle initial. Names are given in their original form.

Contact Details

The postal address or the e-mail address of the author (usually of the first one if there are more Authors) is given in the footnote at the bottom of the first page.

Type of Articles

Classification of articles is a duty of the editorial staff and is of special importance. Referees and the members of the editorial staff, or section editors, can propose a category, but the editor-in-chief has the sole responsibility for their classification. Journal articles are classified as follows:

Scientific articles:

1. Original scientific paper (giving the previously unpublished results of the author's own research based on management methods).

2. Survey paper (giving an original, detailed and critical view of a research problem or an area to which the author has made a contribution visible through his self-citation);

3. Short or preliminary communication (original management paper of full format but of a smaller extent or of a preliminary character);

4. Scientific critique or forum (discussion on a particular scientific topic, based exclusively on management argumentation) and commentaries. Exceptionally, in particular areas, a scientific paper in the Journal can be in a form of a monograph or a critical edition of scientific data (historical, archival, lexicographic, bibliographic, data survey, etc.) which were unknown or hardly accessible for scientific research.

Professional articles:

1. Professional paper (contribution offering experience useful for improvement of professional practice but not necessarily based on scientific methods);
2. Informative contribution (editorial, commentary, etc.);
3. Review (of a book, software, case study, scientific event, etc.)

Language

The article should be in English. The grammar and style of the article should be of good quality. The systematized text should be without abbreviations (except standard ones). All measurements must be in SI units. The sequence of formulae is denoted in Arabic numerals in parentheses on the right-hand side.

Abstract and Summary

An abstract is a concise informative presentation of the article content for fast and accurate Evaluation of its relevance. It is both in the Editorial Office's and the author's best interest for an abstract to contain terms often used for indexing and article search. The abstract describes the purpose of the study and the methods, outlines the findings and state the conclusions. A 100- to 250-Word abstract should be placed between the title and the keywords with the body text to follow. Besides an abstract are advised to have a summary in English, at the end of the article, after the Reference list. The summary should be structured and long up to 1/10 of the article length (it is more extensive than the abstract).

Keywords

Keywords are terms or phrases showing adequately the article content for indexing and search purposes. They should be allocated heaving in mind widely accepted international sources (index, dictionary or thesaurus), such as the Web of Science keyword list for science in general. The higher their usage frequency is the better. Up to 10 keywords immediately follow the abstract and the summary, in respective languages.

Acknowledgements

The name and the number of the project or programmed within which the article was realized is given in a separate note at the bottom of the first page together with the name of the institution which financially supported the project or programmed.

Tables and Illustrations

All the captions should be in the original language as well as in English, together with the texts in illustrations if possible. Tables are typed in the same style as the text and are denoted by numerals at the top. Photographs and drawings, placed appropriately in the text, should be clear, precise and suitable for reproduction. Drawings should be created in Word or Corel.

Citation in the Text

Citation in the text must be uniform. When citing references in the text, use the reference number set in square brackets from the Reference list at the end of the article.

Footnotes

Footnotes are given at the bottom of the page with the text they refer to. They can contain less relevant details, additional explanations or used sources (e.g. scientific material, manuals). They cannot replace the cited literature.
The article should be accompanied with a cover letter with the information about the author(s): surname, middle initial, first name, and citizen personal number, rank, title, e-mail address, and affiliation address, home address including municipality, phone number in the office and at home (or a mobile phone number). The cover letter should state the type of the article and tell which illustrations are original and which are not.

Note