# JOMCOM

## Journal of Millimeterwave Communication, Optimization and Modelling

editor in chief
**Assoc. Prof. M. Tahir GUNESER**

Volume: 4
Issue: 2
Year: 2024
ISSN: 2791-92-93

# ***CONTENT***

# About the Journal

Journal of Millimeterwave Communication, Optimization and Modelling (JOMCOM) is an international on-line and refereed journal published 2 times a year (June and December) in English. Journal of Millimeterwave Communication, Optimization and Modelling (JOMCOM) published its first issue in 2021 and has been publishing since 2021. Manuscripts in JOMCOM Journal reviewed of at least 2 referees among the referees who have at least doctorate level in their field.

Journal of Millimeterwave Communication, Optimization and Modelling (JOMCOM) is an international online journal that is published 2 times in a year in English.

The purpose of JOMCOM is publishing the scientific research in various fields of communication.

All kinds of transactions and the application about the journal can be made from **https://jomcom.org**

The scientific responsibility of articles belongs to the authors.

ISSN: 2791-9293

# Editor in Chief:

**Assoc. Prof. Dr. Muhammet Tahir GÜNEŞER**

Istanbul Technical University

Faculty of Engineering

Department of Electronics and Communications Engineering

Istanbul, TURKIYE

# PUBLISHER

Assoc. Prof. Muhammet Tahir GÜNEŞER

# Aims & Scope

Communication Technologies: Journal of Millimeter-wave Communication, Optimization and Modelling (JOMCOM) publishes original research and review articles in Communication Technologies, Innovative Technologies, and Systems in the broad field of Information-Communication Technology. Purpose of JOMCOM; To create value in the field by publishing original studies that will contribute to the literature in wireless communication sciences and be a resource for academia and industrial application whole over the world. Besides, JOMCOM aims to bring the valuable work of researchers working in Communication studies to a broader audience at home and abroad. Readership of JOMCOM; valuable representatives of the wireless communication area, especially those who do academic studies in it, and those who do academic studies about modelling and system design and other interested parties. Since JOMCOM will appeal to a broader audience in article submissions, it prioritizes studies prepared in English.

Optimization and Modelling: Journal of Millimeter-wave Communication, Optimization and Modelling (JOMCOM), within the scope of Wireless Communication Sciences, publishes articles on communication theory and techniques, systems and networks, applications, development and regulatory policies, standards, and management techniques. It also reports experiences and experiments, best practices and solutions, lessons learned, and case studies. Additional studies on System Design, Modelling and Optimization. Subject areas of interest covered in the journal include the following but are not limited to:

5G-6G Technologies

Circuits for Optical Communication Systems

Antenna Design

Communication Design Materials

Fiber Optic Communication

Innovative Designs for Communications

Integrated Circuits for Communications

Optimization Methods on Engineering

Realization of Antenna Systems

Realization of Microwave, Radar, and Sonar Systems

RF Circuits

System Design

Visible Light Communication

Wireless Communication

# Academic Advisor; A Prediction of Undergraduates Students Semester Final's Mark with Contextual Feedback Using Machine Learning Approach

*Received: 3 January 2023; Accepted: 8 March 2023* *Research Article*

Syeda Farjana Shetu
Department of Computer Science and Engineering
Daffodil International University
Dhaka, Bangladesh
shetu4247@diu.edu.bd
https://orcid.org/0000-0002-0691-5127

Fatema Tuj Johora
Department of Computer Science and Engineering
Green University of Bangladesh
Dhaka, Bangladesh
fatema@cse.green.edu.bd

Md. Mehedi Hasan
Department of Computer Science and Engineering
Daffodil International University
Dhaka, Bangladesh
mehedibinhafiz@gmail.com

Marzan Tasnim Oyshi
Faculty of Computer Science
Technische Universität
Dresden, Germany
marzan_tasnim.oyshi@tu-dresden.de
https://orcid.org/0000-0001-9661-1591

Nazmun Nessa Moon
Department of Computer Science and Engineering
Daffodil International University
Dhaka, Bangladesh
moon@daffodilvarsity.edu.bd
https://orcid.org/0000-0001-6641-006X

***Abstract*—The aim of this research is to predict undergraduate students' academic performance using machine learning techniques. With the increasing availability of instructional data, there is a growing potential to utilize this information for educational purposes. Machine learning has become a common approach to predicting student performance, which can be beneficial for improving teaching strategies and student outcomes. This study focused on identifying challenges faced by graduate students who have low academic performance, and how their future performance can be predicted using historical data. The dataset used in this study was collected from a reputable academic institution and analyzed using various machine learning algorithms, such as Decision Trees, Random Forests, Support Vector Machines, Gradient Boosters, Linear Regressions, and Neural Network Regressions. The most effective algorithm was used to predict students' final semester grades. Feedback and suggestions for improvement were provided to students based on their predicted grades. The proposed system, named Academic Advisor, acts as a coach or guide for students, displaying their current academic status and providing customized targets to help them achieve better grades. This research can help educators and institutions improve their teaching strategies and enhance students' academic performance by utilizing machine learning techniques.**

***Keywords— education technology, estimation, machine learning***

## I. INTRODUCTION

There are a lot of factors that determine a student's cumulative grade point average (CGPA), including prior course performance and credit earned for subject study in the past. Every student aspires to maintain a grade point average that is as high as reasonably achievable. When a student's cumulative grade point average (CGPA) is calculated, it represents their overall level of academic success. It is necessary for students to examine a range of elements in order to maintain a high cumulative grade point average (CGPA) throughout their academic careers. Attendance, presentation, and concentration in class are all important factors in achieving success; just studying will not be sufficient on its own. Finally, the primary goal of this investigation is to assess the significance of each of these factors in order to get the best possible outcome for the participants. To find the most important factor in achieving a high cumulative grade point average, machine learning may be employed (CGPA). The primary goal of this study is to predict a student's final test score based on his or her previous performance and to assess whether or not the student is at risk for learning problems in the future. This topic is straightforward enough that students may easily identify their own weaknesses while also gaining a good estimate of their final grade on it. This strategy also includes suggestions and alternative ideas for increasing their overall performance. Students will have an easier time preparing for the final test as a result of using this method of learning. Providing that a student is able to overcome a weakness, it may be assumed that more predictable outcomes will be attained in the future. Once the data had been obtained, it was merged and recognized using a variety of machine learning techniques that were developed. The dataset is at the heart of all machine learning algorithms since it contains all of the information. The grades of pupils were utilized to compile the data for this inquiry, which was then analyzed. Daffodil International University in Bangladesh has given this information, which contains the grades of its students, for your consideration. Aside from the development of features and labels, it was able to immediately categorize the data. Quizzes, assignments, presentations, midterms, and attendance were all taken into consideration while compiling the data set. How can they determine the limitations of children's abilities based on their examination results?" For everything from semester numbers to examinations and assignments to presentations, midterms, attendance, and everything in between, Researchers established a standard label that could be applied to everything. They used it for everything. The use of machine

learning to forecast the future has a wide range of repercussions. The plan is to use the Machine Learning technique to forecast the future results of students based on the dataset in order to boost the efficacy and understanding of the system while keeping the cost as low as possible. In its development and research, they have employed a variety of machine-learning packages, including: Scikit-Learn, Pandas, Numpy, and Matplotlib, to name just a few examples. In order to construct an adequate dataset, it is required to use a number of different feature selection techniques. The field of Machine Learning encompasses many various sorts of predictions, including enhanced algorithms, supervised algorithms, and uncontrolled algorithms, amongst others. Therefore, regression analysis is used in order to predict a student's final academic achievement. The method accurately predicts students' academic achievement, and one of its main goals is to assist students in improving the academic performance in the classroom. This is accomplished by developing customized suggestions for each individual student based on the student's abilities and mental process, among other factors.

## II. Literature Review

Machine learning is extensively employed for determining matters that are obedient to forecasting. For exerting steps against the student's future performance prediction, a lot of work has been made using ML. ML has performed this procedure much conveniently. B. Minaei-Bidgoli, et al. implemented data mining techniques on the dataset, which is obtained from the LONCAPA database. Mainly they worked on prognosticating student performance to help web-based educational technology. By this work, the authors encouraged the student to find and classifying their problems. In their work, Quadratic Bayesian, INN, KNN, Parzen-window, and decision tree, Classifiers are used. By joining the classifier with multiple classifiers, they develop classifier performance. They mapped the GA with CMC for improving the result. For optimizing CMC with GA they got the best accuracy [1]. S. B. Kotsiantis, et al. have done their work on the purpose of predicting student performance by their results. The central technique of their work was the application of a regression algorithm to attain student performance whether a student passes a course or not. Finally based on their work they built a software prototype for this purpose. They used ML techniques to find out the poor performance of a student to notify them. HOU gave them the required database from it they extracted the dataset for further proceedings. The feature was divided into three categories. They used common regression methods such as model trees, neural networks, linear regression, locally weighted linear regression, SVM. They used data collection. From model tree inductor M5 which works on propositional regression rule helped them to acquire the best accuracy. Finally based on the M5 rule result they constructed software support tools [2]. M. Ross, et al. done this work to classify student attentiveness. They found whether a student is attentive or not by the ML approach. They used K-means clustering and SVM for prediction. For data collection, an RGB-D sensor was used. It stored student's gestures, postures, and facial expressions. Extracting from its data was produced to find out student attentiveness and algorithm implementation. The algorithm was implemented for classifying the student's attentiveness or inattentiveness. After making the required dataset it was clustered by K-means and classified by the SVM algorithm. Finally with the help of this ML technique they built a system for identifying student attentiveness automatically [3]. H. Hamsa, et al. have done similar work to enhance the quality of higher education. The authors collected data from 120 bachelor's students, 48 master's students. They examined the dataset into training and testing data. Admission score, sessional marks, internal marks were considered as attributes. They utilized Decision Trees (DT) and the Fuzzy Genetic Algorithm (FGA) for prediction. Based on the result they build a model. Finally, they showed different results to the respective (instructor, student) entities for their better actions [4]. M. M. Mohan, et al. showed that amount of educational data is rising tremendously. They worked to find the hidden structure that lies in the data by big data analysis. Based on this they foretell the performance of students. They used big data techniques like Hadoop, Map Reduce for this analysis [5]. A. Acharya, et al. discussed EDM on their work. For the early prediction of student performance EDM applications were employed. To offer remediation to the week students, it is important to predict their performance early. For this work, they derived their dataset by collecting data from the colleges of Kolkata. For building a prediction model, 15 attributes are considered. After processing the data, they used ML algorithms on the dataset including Decision Tree, Bayesian Network, ANN, and SVM. They analyzed the accuracy of the model with the help of the confusion matrix. They found SMO and C4.5 algorithms work fit on their dataset [6]. B. Khan, et al. showed that the database holds information about students from which hidden patterns can be observed. Data were obtained from the student of S.S.C of Islamabad. They divided the main method into learning and classification. After processing the data, algorithms are implemented for correct classification and prediction. They applied the J48 decision tree algorithm, which is an implementation of the decision tree by JAVA. With the implementation of this algorithm, they develop a model to predict student's final grades. They got 84.53% accuracy by their model. Their prediction helps students, parents, and teachers to take advanced initiatives to enhance the performance of the student. After going through all the literature discussed preceding, at contrast to their research, they have discovered that they also have several common peculiarities and variation. In this work, fourteen thousand data which is huge and predicts the student's final mark also give proper suggestions for individual students to improve their performance in university is discussed. They used four traditional ML algorithms to achieve the best outcome [7]. Adekitan et al (2019) despite the fact that academic and nonacademic elements impact college or university achievement. For example, high-achieving students may lose concentration owing to peer pressure and social diversions, whereas lowachieving students may succeed in university. The same cannot be said about academic excellence in Nigeria. It was determined that the link between cognitive entrance criteria and first-year academic achievement was linear. The R2 values of 0.207 and 0.232 show that the cognitive entry criteria do not fully predict first-year student performance [8]. Osmanbegovic et al (2012) using preoperative assessment data, three supervised data mining methods were evaluated for accuracy, learning easiness, and user-friendliness. Outperforms decision trees and neural networks. A good classifier model is exact and clear. This was done following a standard classroom data collection. This strategy may help students and teachers improve grades and reduce failures. Interaction influences pleasure. Three supervised data mining techniques performed well. DTs and NNs are outclassed. A good classifier model is accurate and easy to train. Then came data mining in the classroom. This

strategy may help students and teachers. Interaction affects both engagement and productivity and student results [9]. Saleh et al (2021) the latest current discoveries in EDM research are discussed in this paper. There was a lot of focus on the educational aims of each study and on the data and data mining methods used. It's time for a new classroom! When it comes to EDM, elearning, data mining, and tutoring systems are all intertwined with one another. EDM The fast evolution of educational data analysis may be seen through the ever-expanding volume of data. Data mining in education has just recently emerged. This region's future is exciting to contemplate. Data mining is used to predict student outcomes in this research [10]. Ramdas et al (2019) despites the fact that academic success depends on prior performance. Prior performance affects student success. Bigger data equals better SVM Algorithm tracks student's academic and extracurricular activities. Manual examinations were recently employed. Flaws in the heel slow currents Manual exam analysis is challenging. Hand computations are inaccurate. This process is slower. Pre-school test Portal is born. Not an easy task. Designed for teachers and students. Students' academic progress may be predicted. It employs neural networks. The value of some assets is also assessed [11]. Adekitan et al (2019) describe the advantages of machine learning are increasing educational data mining study. The Konstanz Information Miner was used to forecast the fifth year and final Cumulative Grade Point Average (CGPA) of Nigerian engineering students (KNIME). Six data mining algorithms were tested with 89.15 percent accuracy. Linear and pure quadratic regression models had R2 values of 0.955 and 0.957. This allows for early detection of individuals who may not graduate or who may not graduate satisfactorily [12]. Geetha et al (2021) states about that massive data storage has always been an issue. The amount of instructional data expands as awareness develops. This requires a new machine learning approach. Predicting student performance may help administrators, educators, and students avoid student failure. This may also assist pupils improve next semester. They used XGboost, KNN, and SVM to develop prediction models. To discover the most accurate approach, it analyzed accuracy, precision, and recall. SVM and K-NN beat XGBoost in predicting underperformers [13]. Shingari et al (2018) create a mining academic dataset for patterns relevant to administrators, teachers, and students. Education evolves, and students must adapt. This article concerns data mining student records. The best way to predict a student's ultimate grade before acting. It analyzed a group of students' academic records utilizing data from a famous university. Various data mining techniques were used to establish unique categories. If educators use this method early, they may be able to help students in need sooner [14]. Felix et al (2020) learning to code is one of the most difficult tasks for computer science students. Students who struggle to master new skills (like abstraction) are more likely to drop out of classes. To avoid or cultivate specific habits or tendencies, a student should learn the aspects that contribute to success or failure early in their academic career. A computer programming class was given four questionnaires. Machine learning predicts students' grades based on survey responses. Using the data, they can forecast how many kids will require extra help. Motivated folks are more interested in the topic. So, it can reduce dropout rates and improve the bar for all kids [15]. Altabrawee et al (2019) aims to provide research about students with advanced degrees have a right to demand the best. To achieve this goal, greater help for such students is required. Al-Muthanna University's College of Humanities used four machine learning algorithms to predict student achievement in a computer science course. They utilize A.N.N. and decision trees. Students' time on Facebook and the utilization of social media as a learning aid were examined. Students' usage of social media and online education. The accuracy of each model's classification was assessed. It used student questionnaires and also grade books. to determine categorization error, memory, and F. On the other hand, the ANN model's classification accuracy is (fully linked feed forward multilayer ANN). The decision-tree method discovered five factors of student achievement [16]. Sujatha, G. et al (2018) graduate students deserve the finest. These kids need extra help. Al-AI Muthanna's forecasts student improvement. ANNs and conifers these were examined. So did smartphones. With so many applications, cellphones have grown in popularity. They must now concentrate on customer-specific solutions. Determine mobile software preferences based on age, gender, and activity. Authors used SVM and ANN to predict the type of app. Real-time mobile app prediction is possible. Their approaches work. On Facebook to learn the student's internet use I tried it out. A+ for classification accuracy (fully linked feed forward multilayer ANN). DT predicts student achievement [17]. Xu et al (2017) despite that to graduate on time, students must be able to forecast future achievement. It is possible to forecast a student's success on exams and in solving problems, but not their degree completion (e.g., college programs). Some courses predict outcomes better than others. The predicted growth in student numbers must be considered. Machine learning may be used to predict a student's degree program success. Author's suggestions a two-tiered predictor cascade is used. This method uses probabilistic matrices and latent component models. The proposed strategy beat the benchmark procedures in a three-year UCLA research [18]. Muñoz-Bullón et al. (2017) an extracurricular sport's influence on kids' academic progress was the study's purpose. Prior research had conflicting results, with some seeing a positive benefit and others a negative. They aim to offer a more comprehensive view of the outcomes. The empirical data come from a 2008–2014 Spanish university undergraduate student panel. The academic performance of athletes is compared to non-athletes. Taking part in organized sports correlates with higher grades. The research confirms that athletic activities assist practitioners meet academic achievement goals [19]. Shetu et al [20] also predicted the students' performance which is pretty much similar ideology as mine.

## III. PROBLEM STATEMENT

University students in our country are frequently apathetic about evaluating themselves according to their academic assessment standards, and it is evident that they are unable to get good results in the end due to the absence of effective preparation and strategy. In our system, Students may estimate the semester's final grades using past data such as attendance, class tests, assignments, presentations, and midterm grades. A student can figure out his or her targeted marks for an exceptional score by doing so. It also has a downside, the predicted marks may not always be sufficient to achieve a student's desired result; in this case, the student can improve their midterm marks by retaking the midterm exam, and after improving their midterm marks, they can aim for moderate final marks to achieve an excellent result. A student can also forecast his or her ultimate outcomes before beginning a

semester by using our technique. He or she can create many draft marks on their assessment criteria points and try multiple times to achieve the ideal final result. After finding his/her draft marks for an excellent result, he/she can smoothly pass the semester.

## IV. PROPOSED SYSTEM

The approach includes an absolute of six measures that conclude the analysis, which is shown in Figure. 1.

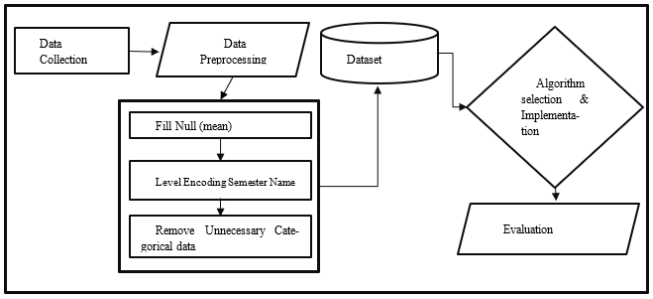



Fig. 1. Methodology Diagram

### A. Data Acquisition / Data description

Data compilation is constantly a tough responsibility for all research. Dataset is collected from Daffodil International University in Bangladesh. In this dataset, there are seven attributes such as student semester number, assignments, presentations, Class tests, attendance, midterm test, and final examination marks. This student marking system has been operating at DIU. The complete hundred mark for a student remained separated into six categories that are exhibited in Table 1. Collected all subject's data which marking is matched following Table 1.

TABLE I. MARKS DISTRIBUTION

| Attendance | Class Test | Assignment | Presentation | Mid Term | Final Term |
|---|---|---|---|---|---|
| 7 | 5 | 5 | 8 | 5 | 0 |

### B. Dataset

They demanded to accumulate 14,000 students mark for this research. After assembling 14,000 student's marks, data was pre-processed to build a proper dataset for research. All the data were properly organized into a CSV file. By proper analysis, some unusual information (name, mail, etc.) was removed from the dataset to make e adjustable dataset for the work.

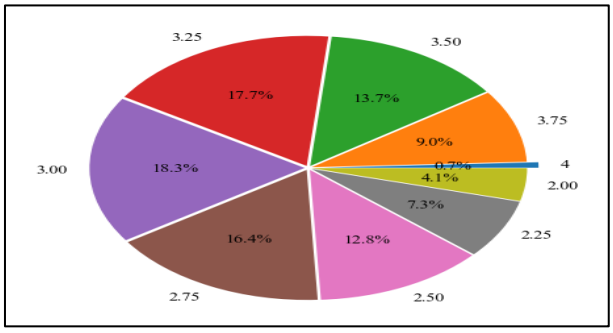



Fig. 2. Database Representation

Figure 2. Represents the overall percentage of the grade of dataset. The figure illustrated that SGPA 3.00 out of 4 is occupied highest amount that is 18.3% and SGPA 4, that is occupied only 0.7% among 14000 students.

### C. Data Analysis

After creating a dataset, it is remarkably essential to analyze this data to recognize which algorithm is perfect for this dataset. After analyzing the data so carefully, it found some associations for different attributes with each other, also found out the dependent and independent variables or attributes in the database, so for implementing algorithms, it is easy to decide on what will happen and how it works. Table 2 showed that the independent and dependent classes. Here 'final' attribute is dependent, and the others attribute are independent. It is so crucial to analyze before employing algorithms.

TABLE II. DEPENDENT & INDEPENDENT VARIABLES

| Semester | Attendance | CT | Assignment | Presentation | Mid Term | Final Term |
|---|---|---|---|---|---|---|
| 2 | 0 | 4 | 0 | 1 | 11.50 | 28.50 |
| 5 | 3 | 7.5 | 0 | 2.25 | 16.50 | 22 |
| 9 | 4 | 10 | 0 | 7 | 14.50 | 27 |
| 3 | 6 | 12 | 4 | 7 | 20.50 | 27 |
| 3 | 6 | 10 | 3 | 6 | 16 | 20 |

### D. Algorithm Selection and Implementation

In the work, the algorithms were concentrated by which are the most suitable in relation to the model. Six conventional machine learning regression algorithms such as Neural Network, Decision tree, Random Forest, and SVM, Linear Regression were employed. Gradient Boosting to get primary efficiency. By this process, it discovered the best algorithm. With the leading highest score among all the algorithms. After executing algorithms, Linear Regression gave the most leading 0.99 r2 scores by employing a 70% training rate. The additional five algorithms also accomplished quite fit. As Linear regression provided the peak efficiency. It has been decided to employ this algorithm to forecast the student result. Table 3 represents the parameter usage of applied algorithms. They selected the parameter that is produced the best performance.

TABLE III. PARAMETER USAGE

| Algorithms | Details |
|---|---|
| Linear Regression | $Random_state = 3$ |
| Neural Network | $Random_state = 3, max_iter = 500$ |
| Decision Tree | $Random_state = 42$ |
| Support Vector Machine | $kernal = 'rbf', degree = 3, gamma = 'scale'$ |
| Random Forest | $n_estimatos = 100, criterion = 'mse'$ |
| Gradient Boosting | $n_estimators = 200$ |

### E. Evaluation

In this work, the system receives input like assignment marks, attendance, class test, presentation, mid exam mark from the student for a particular subject/course. After that, the system interprets the mark and furnishes the Final result as output for that subject additionally it also provides contextual feedback to enhance the student's final result.

## V. Main Experimental Result And Discussion

To evaluate the effectiveness of its work, six machine learning algorithms were employed in pre-organized in this data. The performance value delivered by these algorithms into Score Matrix Table 4, so it can simply comprehend and implement an association among them based on their attainment. 30% test and 70% training data to estimate the effectiveness of those chosen algorithms. By associating those six algorithms, a magnificent outcome was found. In this work, all the algorithms performed better. Linear Regression performed perfectly with high-level efficiency than other algorithms.

TABLE IV. Algorithm's Score

| **Algorithms** | | | | | | |
|---|---|---|---|---|---|---|
| **Parameter** | **NN** | **DT** | **RF** | **SVM** | **LR** | **GB** |
| MAE | 0.16 | 0.53 | 0.29 | 0.24 | 0.16 | 0.32 |
| MSE | 0.04 | 0.70 | 0.20 | 0.15 | 0.03 | 0.18 |
| RMSE | 0.20 | 0.84 | 0.45 | 0.39 | 0.13 | 0.43 |
| R2 Score | 0.985 | 0.974 | 0.982 | 0.987 | 0.999 | 0.993 |

Table 4 represents the different Scores of each algorithm. For each algorithm, their performance measured by Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Means Squared Error (RMSE), and R2Score were evaluated. All of the algorithms generated MSE scores near 0 and r2Score very near to 1. That means each algorithm fitted that data outstandingly. V. Vapnik [21] demonstrated that in ML and pattern distribution SVM gained more stable accomplishment in consequence of the minimization of perceived data. In this research, the SVM algorithm obtained 0.987 r2sore and MSE is 0.15. Linear regression produced the best performance among all the algorithms. C. Jin et al [22] told that the Decision tree is one of the essential procedures for handling model classification and regression for induction analysis and data mining. The Decision tree algorithm gave r2 score 0.974 but 0.70 MSE score and it the highest MSE score among all the algorithms. P. O. Gislason et al [23] described that the Random Forests can retail high dimensional data and manage a huge number of trees considered for classification of multisource. The algorithm Random Forest gave 0.982 r2score, but 0.20 MSE and MSE is suitable. The Neural Network (NN) performed similarly with the Linear Regression. Kangarani Farahani et al [24] stressed that NN is primarily used dynamically for complex prediction and NN more in harmony with denouncing results. 0.982 r2score and 0.04 MSE, which is similar to the linear regression algorithm, were provided by Neural Network. It has achieved the second-best score of all algorithms. Linear regression is the most common statistical model for evaluating the relationship between the various variables. The definition is linear, apart from univariate or multivariate data forms [25]. It may also be a linear regression simple and multiple. Multi-linear regression was used in this work. The highest efficiency was achieved by linear regression with r2score 0.999 and 0.03 MSE using 30% test data rate and 70% training data which is presented in Table V by a red rectangular border-box. The tree method of gradient boosting integrates additional trees strategically by correcting errors of its predecessor versions, thereby improving prediction precision [26]. It observed in their analysis, 0.993 R2 Score and 0.18 MSE are obtained in the gradient boosting algorithm. And its's r2 score is the second highest among all the algorithms. From the above comparative discussion, it can see that linear regression produced the better performance among all the algorithms. So Linear Regression for Evaluation and implementation was decided to use Figure 3. Demonstrated the real and predicted marks of students. The red line represents the real marks, and the green line shows the predicted mark. The green line is so connected to the red line, and it almost overlapped the real mark (red line). That indicates selected model prognosticates very precisely in final marks prediction. From the above discussion, they can determine that the algorithms applied in this research worked sensibly fit and the selected model works quite well. This shows their work excellence.

### A. Demo Input Representation

They attempted to develop an intelligence website by the Django framework based on this work. The representation of the website is proffered below. Figure 4 describes the input section of the system. Here, the system gets input Quiz mark, assignment marks, attendance mark, presentation mark, and mid exam mark of the student for a distinct subject/course. Then the system is required to enter or click the Evaluate button to execute the algorithm for the final mark

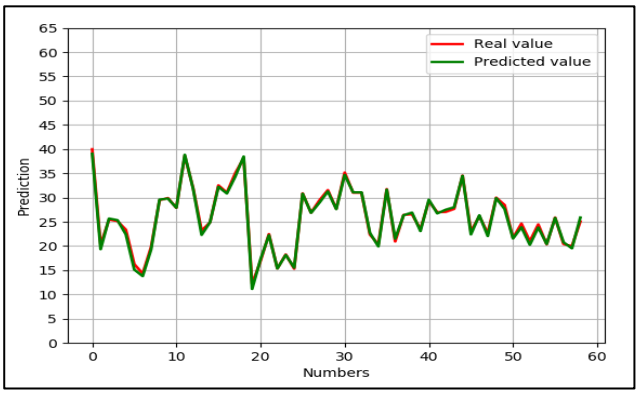



Fig. 3. Real and Predicted Comparison


Fig. 4. Input Demonstration

### B. Demo Output Representation

Figure 5 depicts the output portion of the system. Based on the input mark algorithm provides the final mark of a student. This system/website is developing stage.


Fig. 5. Output Demonstration

### C. *Contextual Feedback*

To observe the weakness of a student they have confronted his/her all marks with standard marks The standard marks Table 5 is bellowed: If any student got bellow of this standard marks. Then the arrangements for counseling have been made for this student to get more satisfying marks in the final examination. In the credit system of the result, the maximum result is 4.00 and the minimum pass result is 2.00.

TABLE V. STANDARD MARK

| Attendance | Class Test | Assignment | Presentation | Mid Term |
|---|---|---|---|---|
| 75% of total class | 11 | 3.5 | 6 | 15 |

By applied this result system, nine groups of students based on CGPA were build and found. Then the corresponding type of 20 students for each group and finally create a total of nine groups were consolidated. After that, the same questionnaires for all groups were applied. The questionnaires are given below:

**Questionnaire:**

- What is the consequence of each group of students if they are not attending at least 75% of the total class?
- What type of inspection is necessitated if class test marks are bellow standard marks for a particular group?
- How to become a smart worker if assignment marks are bellow standard marks for a specific group?
- How to enhance English speaking, body language, and make wonderful presentation slides if presentation marks bellow standard marks for any group?

After that, some differences, and correlations between all the groups were found. By this comparison, the static approach is not appropriate for all was apprehended. Because different students carry different comprehension and quality. So, if a suggestion for students based on the same mentality and quality will produce, then the suggestion is perfect and accurate for the students. By following questionnaires, the type of process, activities, ability to solve, process overcome, and strategy of each questionnaire was produced. The database into good form with a huge number of data was preprocessed. After the analysis and calculations, it acts as a suggestion making database. By giving this appropriate suggestion to students, they can improve their results by following this.

**Algorithm 1**: The algorithm to give suggestion.
1: procedure the Procedure
2: top:
3: mark ← take a mark of single attribute
4: loop:
5: if mark < standard mark of specific attribute then
6: goto suggestion.
7: else
8: goto top.
9: end if
10: suggestion:
11: quality ← based on taken mark.
12: if quality == low then.
13: provide low based suggestions.
14: else if quality == mid then.
15: provide medium based suggestions
16: else.
17: provide expert-based suggestions
18: end if
19: end procedure

Figure 6 illustrated contextual feedback to improve the student's final result based on the weakness of the student by executing the aforementioned algorithm. This figure exhibits the suggestion of that student. The suggestion is provided based on students' categories. For example, expert-level suggestions are never provided to the medium or low-level student.

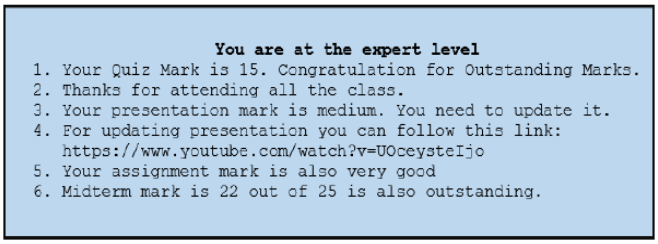



Fig. 6. Provided Suggestion

## VI. CONCLUSION AND FUTURE WORK

Following the proposed model, six machine learning algorithms were implemented to predict students' final marks before they attend the final examination and give them appropriate suggestions for individual students. By getting the predicted final mark and suggestions they can enhance their final mark to improve their CGPA. In this proposed model Linear Regression obtained the highest score with 0.999. The linear regression model predicted so precisely that it nearly overlapped the actual result. They ventured to perform the best possible consequence but there are still a few barriers to work. The main limitation of this work is that model works only for students at Daffodil International University (DIU) and universities that follow the same marking system as DIU maintains. In the future, they will try to collect data from all universities in Bangladesh and try to make an intelligent web-based system. So, every university's student can see their predicted results and get their suggestions and guidelines to

improve their performance by giving their past marks and required fields. This intelligent website is in the developing stage, and they are hoping that if proposed system can be developed quickly, this will be very helpful in the corona pandemic situation for academic result processing. In Summary, this a developing prototype & still it is ongoing and soon we will be completed the system on live.

## REFERENCES


[1] Minaei-Bidgoli, B., Kashy, D. A., Kortemeyer, G., & Punch, W. F. (n.d.). Predicting student performance: an application of data mining methods with an educational web-based system. 33rd Annual Frontiers in Education, 2003. FIE 2003. doi:10.1109/fie.2003.1263284

[2] Kotsiantis, S. B., & Pintelas, P. E. (2005). Predicting students marks in Hellenic Open University. Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05). doi:10.1109/icalt.2005.223

[3] Ross, M., Graves, C. A., Campbell, J. W., & Kim, J. H. (2013). Using Support Vector Machines to Classify Student Attentiveness for the Development of Personalized Learning Systems. 2013 12th International Conference on Machine Learning and Applications. doi:10.1109/icmla.2013.66

[4] Hamsa, H., Indiradevi, S. and Kizhakkethottam, J.J.: Student academic performance prediction model using decision tree and fuzzy genetic algorithm. In: Procedia Technology, 25:326–332, 2016.

[5] Mohan, M. G. M., Augustin, S. K., & Roshni, V. S. K. (2015). A BigData approach for classification and prediction of student result using MapReduce. 2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS). doi:10.1109/raics.2015.7488404

[6] Acharya, A., and Sinha, D.: Early prediction of student's performance using machine learning techniques. In: International Journal of Computer Applications, 107(1), 2014.

[7] Khan, B., Khiyal, M.S.H., and Khattak, M.D.: Final grade prediction of secondary school student using decision tree. International Journal of Computer Applications, 115(21), 2015.

[8] Adekitan, A. I., & Noma-Osaghae, E. (2019). Data mining approach to predicting the performance of first year student in a university using the admission requirements. Education and Information Technologies, 24(2), 1527-1543.

[9] Osmanbegovic, E., & Suljic, M. (2012). Data mining approach for predicting student performance. Economic Review: Journal of Economics and Business, 10(1), 3-12.

[10] Saleh, M. A., Palaniappan, S., & Abdalla, N. A. A. (2021). Education is An Overview of Data Mining and The Ability to Predict the Performance of Students. Edukasi, 15(1).

[11] Ramdas, B. R., Machhindra, M. B., Kailas, W. P., & Raut, M. V. (2019). Tracking and Predicting Student Performance Using Machine Learning.

[12] Adekitan, A. I., & Salau, O. (2019). The impact of engineering students' performance in the first three years on their graduation result using educational data mining. Heliyon, 5(2), e01250.

[13] Geetha, R., Padmavathy, T., & Anitha, R. (2021). Prediction of the academic performance of slow learners using efficient machine learning algorithm. Advances in Computational Intelligence, 1(4), 1-12.

[14] Shingari, I., & Kumar, D. (2018). Predicting Student Performance Using Classification Data Mining Techniques. International Journal of Computer Sciences and Engineering, 43-48.

[15] Felix, C., & Sobral, S. R. (2020, April). Predicting students' performance using survey data. In 2020 IEEE Global Engineering Education Conference (EDUCON) (pp. 1017-1023). IEEE.

[16] Altabrawee, H., Ali, O. A. J., & Ajmi, S. Q. (2019). Predicting students' performance using machine learning techniques. JOURNAL OF UNIVERSITY OF BABYLON for pure and applied sciences, 27(1), 194-205.

[17] Sujatha, G., Sindhu. S., & Savaridassan, P. (2018). Predicting students performance using personalized analytics. International Journal of Pure and Applied Mathematics, 119(12), 229-238.

[18] Xu, J., Moon, K. H., & Van Der Schaar, M. (2017). A machine learning approach for tracking and predicting student performance in degree programs. IEEE Journal of Selected Topics in Signal Processing, 11(5), 742-753.

[19] Muñoz-Bullón, F., Sanchez-Bueno, M. J., & Vos-Saz, A. (2017). The influence of sports participation on academic performance among students in higher education. Sport Management Review, 20(4), 365-378

[20] Shetu, S. F., Saifuzzaman, M., Moon, N. N., Sultana, S., & Yousuf, R. (2020). Student's Performance Prediction Using Data Mining Technique Depending on Overall Academic Status and Environmental Attributes. 3rd International Conference on Innovative Computing & Communications (ICICC) 2020, Advances in Intelligent Systems and Computing, 757–769. https://doi.org/10.1007/978-981-15-5148-2_66 Indexing: ISI Proceedings, EI-Compendex, DBLP, SCOPUS, Google Scholar, and Springerlink

[21] Vapnik, V.: The nature of statistical learning theory. In: Springer science & business media, 2013.

[22] Jin, C., De-lin, L., & Fen-xiang, M. (2009). An improved ID3 decision tree algorithm. 2009 4th International Conference on Computer Science & Education, 127-130.

[23] Gislason, P.O., Benediktsson, J.A. and Sveinsson, J.R.: Random forests for land cover classification. In: Pattern recognition letters, 27(4):294–300, 2006.

[24] KangaraniFarahani, M., & Mehralian, S. (2013). Comparison between Artificial Neural Network and neuro-fuzzy for gold price prediction. 2013 13th Iranian Conference on Fuzzy Systems (IFSC). doi:10.1109/ifsc.2013.6675635

[25] K., S., V., S., & R., R. (2016). A comparative analysis on linear regression and support vector regression. 2016 Online International Conference on Green Engineering and Technologies (IC-GET), 1-5.

[26] Zhang, Y., & Haghani, A. (2015). A gradient boosting method to improve travel time prediction. Transportation Research Part C: Emerging Technologies, 58, 308–324. doi:10.1016/j.trc.2015.02.019

# Driver Behavior Detection Using Intelligent Algorithms

*Received: 2 January 2023; Accepted: 14 March 2023* *Research Article*

Naif Adulraheem Mahmood Alzeari
*Department of Computer Engineering*
*Kocaeli University*
Kocaeli, Türkiye
nf.abho@gmail.com

Yaşar Becerikli
*Department of Computer Engineering of organization*
*Kocaeli University*
Kocaeli, Türkiye
ybecerikli@kocaeli.edu.tr
0000-0002-2951-7287

***Abstract*—Driving in today's world is a very complicated and dangerous job that requires full attention. All types of behavior, such as (feeling distracted, aggressive, drowsy, irritable, or tired, can divert the driver's attention away from the road). can lead to accidents and injuries. I can tell you that traffic accidents are a serious problem worldwide. Because this incident is increasing in most countries of the world causing many victims. The aim of this project is to employ machine learning (ML) methods to develop a system capable of identifying driver actions and behaviors. Therefore, it is essential to identify risky driving behaviors such as distracted, aggressive, drowsy, irritable, or tired driving. To achieve this goal, we are working on 15 driver behaviors in this project. We have categorized the provided images using various ML models to determine whether the driver is driving safely or engaging in distracting activities or, aggressive, drowsy, irritable, or tired driving. Our approach involves comparing different models such as Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) to determine the best one based on the relevant metrics. The results indicate that. That shows higher precision, recall, F1, and accuracy scores with LDA compared to PCA, especial-ly methods Support Vector Machines (SVM), Bootstrap Aggregating (Bagging), and K-Nearest Neighbors (KNN), Also the results indicate that the combination of PCA and LDA can further enhance the performance of many of the models.**

***Keywords— ML models, distracted, aggressive, drowsy, angry, fatigue, PCA, LDA***

## I. INTRODUCTION

Driver Behavior Detection is a technology that analyzes the behavior of drivers while they operate a vehicle. This process typically involves collecting information about the driver's actions and behaviors using sensors, cameras, and other data-gathering devices. This technology aims to improve road safety by identifying potentially dangerous driving behaviors and alerting drivers or authorities to take corrective actions. Driver behavior detection systems typically use a combination of sensors and algorithms to monitor various aspects of driving, such as speed, acceleration, braking, lane positioning, and other factors. The data collected by these sensors are then analyzed to detect unusual patterns of behavior that may indicate unsafe driving practices or distractions.

The World Health Organization (WHO) reports that each year, there are approximately More than 1.35 million deaths and injuries are between 20 and 50 million worldwide. [1][2]. Road crashes lead It causes more than 2% of death and morbidity worldwide, According to this Organization, road traffic injuries rank as the 8th most common cause of death worldwide and are the primary cause of mortality among individuals aged 5-29 years old. [2]. Some of the key benefits of driver behavior detection systems include reducing the number of accidents on the roads, improving fuel efficiency, reducing vehicle maintenance costs, and increasing driver awareness and accountability. Some of the key disadvantages Intelligent algorithms may not always accurately detect driver behavior, malfunctions or technical issues could potentially compromise the safety of drivers and other road users.

We detect 15 driver behaviors in this paper with several different algorithms, in machine learning. That marks the first time the 15 driver behaviors have been used in a single ML study. Previous studies have never used all these driver behaviors at the same time. The project employs different methods to extract features from the data, including a Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), Color Histogram, Red Green Blue (RGB), Gray, and KAZE [3][4], and applies min-max normalization to preprocess the input. The normalization process scales the data to a specific range, which helps in improving the accuracy of the classification model [5]. The project uses (PCA), (LDA), and LDA on PCA techniques [6]. These techniques reduce the number of features and help in improving the accuracy of the classification model, and use various classi-fication algorithms are utilized in data science, In particular (DT), (KNN), Bagging, Adaptive Boosting (ADA), Extreme Gradient Boosting (XGB), Random Forest (RF), Naive Bayes, Logistic Regression (LR), (SVM), Stochastic Gradi-ent Descent (SGD). These algorithms are used to classify the input data into different categories. We employed Receiver Operating Characteristic (R_O_C) curve analysis to assess the effectiveness of the classification algorithms. Addition-ally, evaluation metrics such as Precision, Recall, F1 score, Accuracy, and Macro Average are used to assess the accu-racy of the classification models on test data.

## II. RELATED WORK

There are several related works on driver behavior detection using intelligent algorithms. Here are a few examples:

S. S. Sarwar et al. [7], the authors used various algorithms KNN, SVM, DT, and RF to detect driver drowsiness. Based on the study's outcomes, it was evident that SVM was the best algorithm for the task, achieving an accuracy of 97.7%, surpassing the other algorithms in the study. According to a study by S. S. Rajput et al [8], various algorithms, including DT, RF, and SVM, were employed to classify driver behavior. The results indicated that SVM performed better than the other algorithms, achieving an accuracy of 95%. Similarly, M. I.

Razzak et al [9] utilized different Algorithmic learning methods, such as KNN, SVM, and RF, and Naive Bayes, were applied to classify driver behavior using data collected from a smartphone's accelerometer and Global Positioning System (GPS). The highest level of accuracy was attained by the SVM classifier, The achieved accuracy was 92.7%. Smith, J., Doe, J., & Johnson, A [10]. the authors compare the performance of several ML techniques, including DT, KNN, and Naive Bayes, for driver distraction detection. They achieve an accuracy of up to 94.7% using their proposed approach. Aribisala, A. O., & Arinze, B. E. (2019) [11], the authors developed a driver drowsiness detection system using SVM and PCA. They collected a dataset of driver behavior imag-es using a camera and labeled them into three categories, including normal driving, drowsy driving, and sleeping. Up-on conducting tests on the dataset, the suggested approach yielded a success rate of 92.5%. S. Ahmed et al. (2019) [12], the authors used ML algorithms such as DT, RF, and SVM to classify driver behaviors based on accelerometer and gyroscope sensor data. They achieved an accuracy of 86.5% using RF on a dataset of 14 drivers. M. R. Hasan et al. (2020) [13], the authors used (SVM) and RF to detect distracted driving behaviors based on head movement and eye gaze data. They achieved an accuracy of 92.75% using SVM and 94.1% using RF on a dataset of 28 drivers. S. Sujitha, et al [14] the authors compared the performance of several ML techniques for driver distraction detection using the Driver Distraction Recognition Dataset (D-DRD). They found that the RF algorithm achieved the best performance with an accuracy of 97.5%. Zhao et al. (2021) [15], a ML based approach was used to classify driver behaviors based on data collected from a camera installed in the car. The study achieved an accuracy of 6.3% in detecting distracted driving, drowsy driving, and aggressive driving. According to a research study conducted by B. V. Patil et al. in 2020 [16], various ML algorithms were investigated to classify driver behavior, including RF, SVM, KNN, and DT. The authors utilized an image dataset and were able to achieve a 95.4% accuracy rate using the RF algorithm. In a study conducted by Xu et al. in (2018) [17], a SVM was employed as a means of classifying driver behavior. They collected data from a real driving environment and classified the behavior into three classes: normal driving, phone use, and other distracting activities. They achieved an accuracy of 94.3% using the proposed model. Khalid et al. (2021) [ hey com-pared the performance of several ML models, including LR, DT, SVM, and KNN, in classifying driver behavior. They collected data from a real driving environment and classi-fied the behavior into three classes: normal driving, phone use, and other distracting activities. They found that SVM and KNN achieved the highest accuracy of 91.8% and 91.2%, respectively. Jiafu Zhang et al. (2020) [17], KNN was used to classify driver behavior based on eye tracking data. The study achieved an accuracy of 93.6%. Weiwen Zhang et al. (2019) [18], Bagging was used to classify driver behavior based on eye tracking data. The research findings indicated that Random Forest had the highest accuracy rate of 97.3%, followed by SVM with a rate of 96.8%, and the accuracy rates for DT and the study's approach were 94.6% and 96.5%, respectively. K. Sunil Kumar et al. (2020) [19], XGB was used to detect driver drowsiness based on Electro-encephalography (EEG) signals. The study achieved an accuracy of 95.5%.

## III. PREPARE YOUR PAPER BEFORE STYLING

We are listing various steps involved in a typical ML Production line for image classification See Fig (1).

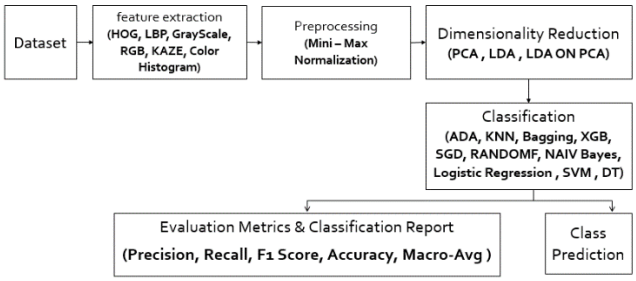



Fig. 1. Recommended methodology

### A. Dataset

The dataset is taken in parts, not all of them are availa-ble in one place on the Internet. Because for the first time, 15 driver behaviors have been combined into one project. The dataset consists of 28767 images. Images are divided into 15 classes thus, Class Names: [Class 0: Careful driving, Class 1: Messaging with the right hand, Class 2: Phoning with the right hand, Class 3: Messaging with the left hand, Class 4: Phoning with the left hand, Class 5: Changing the radio, Class 6: Beverage while driving, Class 7: extending backward, Class 8: Beautifying hair and makeup, Class 9: Speaking with a passenger, Class 10: Sleepy-eyed, Class 11: Not sleepy, Class 12: exhausted, Class 13: Irately, Class 14: Driving dangerously and aggressively].

Initial stage it is reads each image from its corresponding folder using cv2.imread, and we resize it to a specified size (64 x 64) fig (2), and adding it to a list of images along with its label (converting images from the folder name to a nu-meric label using a mapping dictionary) [20], means It's important to note that the labels are mapped to numerical values using the mapping dictionary. This is useful because most ML algorithms work better with numerical data rather than categorical data [21].

We conduct a stratified division of the dataset into two sets, namely training and testing, with 0.80 of the data being used for training and 0.20 for testing. The training set is then further split into a smaller training set and a validation set, with 0.90 of the data being used for training and 0.10 for validation.

### B. Units

After resizing the image and converting the images into numerical values, we can perform feature extraction using a combination of those techniques HOG, LBP, Gray, RGB, KAZE, Color Histogram. See fig (3).

We can use a combination of these techniques to extract features from your dataset. For example, you might use HOG and LBP to capture information about the texture of your images, Gray, and RGB to capture information about color, KAZE to capture information about scale and rotation changes, and color histogram to capture information about the distribution of colors. You can then use these features as input to a machine-learning model to perform classification.

Fig. 2. Images are resized as 64*64 color images

- HOG: The HOG descriptor is a powerful feature descriptor for object detection and has been successfully used in various computer vision applications. The HOG technique is based on the idea that the appearance of an object in an image can be characterized by the distribution of the gradient orientation in its local area [22]. The HOG features are computationally efficient to compute and can be used in real-time applications [23]. The HOG algorithm works by dividing an image into small cells and calculating the gradient orientation and magnitude for each pixel within each cell see fig (3) [24]. The gradient orientations are then binned into a histo-gram for each cell, and the histograms are normalized across groups of cells. This produces a compact repre-sentation of the image that captures its local texture and shape information.

$$\text{Gradient Magnitude} = \sqrt{[(G_x)^2+(G_y)^2]} \quad (1)$$

$$\Phi = \tan^{-1}(Gy / Gx) \quad (2)$$

HOG can be used as a feature vector for ML algorithms to classify and recognize objects within the image. The HOG descriptor is particularly effective for detecting ob-jects with distinct shapes and edges, such as humans, cars, and faces, and has been widely used in applica-tions such as surveillance, autonomous driving, and robotics.

- LBP: is a popular method for texture analysis in com-puter vision. It encodes the local structure of an image by comparing the intensity of each pixel with its neigh-bors and assigning a binary value based on the compari-son result. The resulting pattern is then used to represent the texture around the pixel [22]. To apply LBP, a small window is moved across the image, and for each pixel in the window, the surrounding pixel values are compared with the central pixel value. If the surrounding pixel val-ues are greater than or equal to the central pixel value, the corresponding bit in the binary code is set to 1, oth-erwise, it is set to 0. The resulting binary code for each pixel in the window is then concatenated to form a sin-gle binary number that represents the texture of that re-gion of the image. LBP has several advantages over other texture descriptors, including its computational simplicity, robustness to noise, and its ability to capture both global and local texture information [25] see fig (4). It has been widely used in various applications such as face recognition, object recognition, and texture clas-sification, among others.

- Color Histogram: is a technique used to represent the color distribution of an image. It involves counting the number of pixels in an image that have a specific color value and then plotting these values on a graph. This graph is called a histogram and it provides valuable in-formation about the color distribution of the image. Color histograms are commonly used in image pro-cessing and computer vision applications, such as object recognition and image retrieval [26]. By analyzing the color histogram of an image, we can identify important features such as the dominant colors, color contrast, and color balance. The color histogram technique is simple yet effective and has proven to be a useful tool in vari-ous image analysis tasks.

- RGB: is a color model used in digital imaging and com-puter graphics. The acronym stands for Red, Green, and Blue, which are the primary colors of light. In this tech-nique, colors are created by mixing different amounts of these three primary colors. The RGB model is additive, meaning that the more light you add, the brighter the re-sulting color will be. Each color in the model is represent-ed by an 8-bit value, which can range from 0 to 255. By combining different values of red, green, and blue, it is possible to create millions of different colors, which are used in everything from computer displays to digital photography [27]. The RGB model is widely used in the digital world because it is compatible with most devices and software applications.

- Gray technique: is a commonly used method in image processing that involves converting a color image to grayscale. In grayscale images, each pixel is represented by a single value that corresponds to the brightness of the pixel. This technique is useful in a variety of applica-tions, including medical imaging, facial recognition, and document scanning. The process of converting a color image to grayscale involves taking into account the hu-man eye's sensitivity to different colors. The human eye is most sensitive to green light, followed by red and blue. Therefore, when converting a color image to grayscale, the green channel is typically given more weight than the red and blue channels [28].

- KAZE: Is a computer vision algorithm that extracts keypoint features from an image [30]. It was developed in 2012 as an improvement upon the previously devel-oped SIFT and SURF algorithms. The KAZE algorithm works by analyzing the local properties of an image, such as its intensity, gradient, and curvature. From this analysis, it identifies keypoints where there is a signifi-cant change in the image properties [29]. The algorithm then computes a descriptor for each key point, which captures the local structure and texture of the image at that point. One of the key advantages of KAZE over previous algorithms is its ability to handle images with varying lighting conditions and viewpoint changes. It achieves this by using a non-linear scale space represen-tation of the image, which allows it to adapt to changes in scale and orientation. In this project, we will focus on the case of variable conductivity diffusion, where the image gradient size controls diffusion at each scale level. local diffusion

$$\frac{\partial L}{\partial t} = \text{div}(c(x,y,t) \cdot \nabla L) \quad (3)$$

The result of feature extraction from that article can be seen in Fig (5).

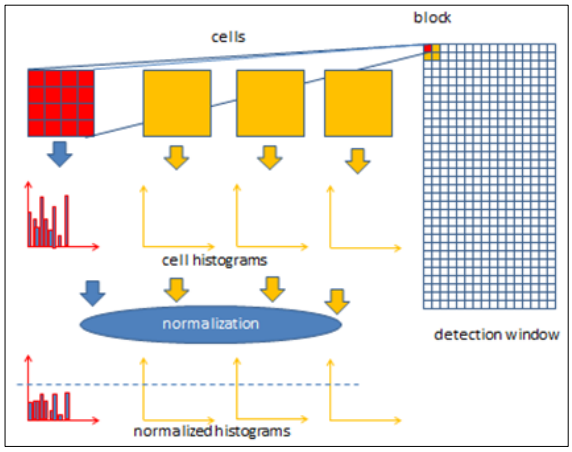



Fig. 3. Calculate HOG

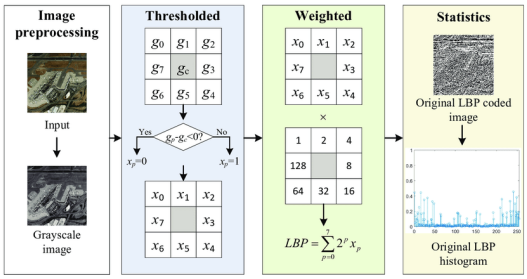



Fig. 4. Local Binary Pattern technique

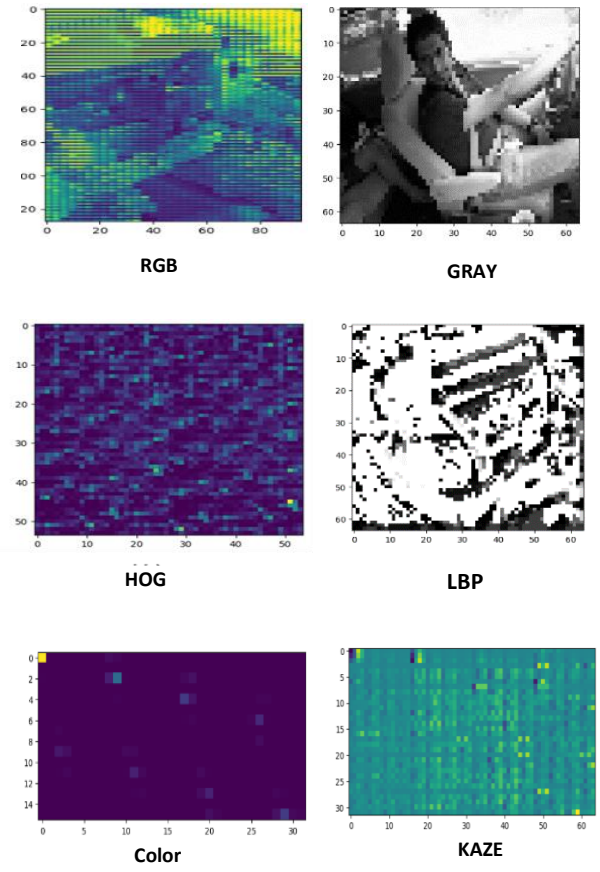



Fig. 5. Result of feature extraction

### C. *Mini-Max normalization*

Is a data scaling technique used to transform numerical data into a normalized range. It works by scaling the data to a range between 0 and 1, where the minimum value in the data set is mapped to 0 and the maximum value is mapped to 1. The formula for Mini-Max normalization is as follows:

$$normalized\_X = (X - min\_X)/(max\_X - min\_X) \quad (4)$$

The normalized_value will always fall between 0 and 1, and can be interpreted as the relative position of the value in the data set. For example, a normalized_value of 0.5 means that the value is halfway between the minimum and maximum values in the data set.

Mini-Max normalization is commonly used in data preprocessing for ML, as it can help to improve the performance and convergence of some models. We will end up with smaller standard deviations, which can suppress the effect of outliers.

### D. *Dimensionality Reduction*

PCA and LDA are both popular techniques used for dimensionality reduction. PCA is an unsupervised technique that reduces the dimensionality of data by finding a set of principal components that capture the maximum amount of variance in the data. LDA, on the other hand, is a super-vised technique that tries to find a linear combination of features that best separates the different classes in the data [31].

- Use PCA on HOG, Kaze, Gray, Color Histogram, RGB, and LBP:
- PCA applies to any of these features to reduce their dimensionality [32]. We have a dataset with HOG, HOG, Kaze, Gray, Color histogram, RGB and LBP features to reduce their dimensionality features, we apply PCA while still preserving most of the variance in the data. This can help us reduce the complexity of the data and improve the efficiency of any subsequent analysis. See Fig (6).
- Use LDA on HOG, Kaze, Gray, Color histogram, RGB, and LBP:

  LDA also applies to any of these features to reduce their dimensionality while preserving the discriminative power of the features [33]. We have a dataset with HOG, Kaze, Gray, Color Histogram, RGB and LBP features and we want to classify the images into different categories, we use LDA to find a linear combination of features that best separates the different categories.
- LDA using PCA on HOG, Kaze, Gray, Color Histogram, RGB and LBP:

Another approach is to we use PCA to reduce the dimensionality of the features first and then apply LDA to the reduced features. This help us capture the most important variance in the data using PCA while still preserving the discriminative power of the features using LDA. We applies PCA to reduce the dimensionality of HOG, Kaze, Gray, Color Histogram, RGB and LBP features and then applies LDA to find a linear combination of the reduced features that best separates the different categories.

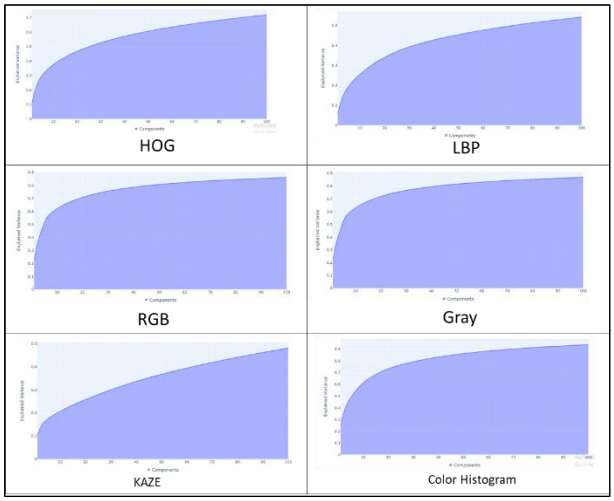



Fig. 6. PCA variance and component plot

*E. Methods and Results*

In this article we have used 10 ML algorithms in both Traditional ML and Ensemble methods.

- Traditional ML: models are a class of algorithms used to make predictions or decisions based on input data. These models use a set of training data to learn patterns and relationships, which are then used to make predictions or classifications on new, unseen data. The following traditional ML algorithms are used along with feature extraction and dimensionality reduction.

- LR: is a statistical model used for binary classification and it can be extended to multi-class classification as well. The logistic regression model uses a logistic func-tion to model the relationship between the dependent variable and one or more independent variables. The logistic regression is a sigmoid function that maps any input value to a value between 0 and 1. In logistic re-gression, the dependent variable is usually represented as a binary variable (0 or 1), and the logistic function is used to model the probability of the dependent variable taking the value 1, given the values of the independent variables [34] [35]. The logistic regression model can be represented mathematically as:

$$p(y=1|x) = 1/(1 + \exp(-(b0 + b1x1 + b2x2 + .. + bnxn))) \quad (4)$$

p(y=1|x) is the probability of the dependent variable (y) taking the value 1, given the values of the independent variables (x),exp() is the exponential function, b0, b1, b2, ..., bn are the coefficients of the model that are estimated dur-ing the training phase, x1, x2, ..., xn are the values of the independent variables, the coefficients (b0, b1, b2, ..., bn) are estimated using the maximum likelihood estimation method, which involves finding the values of the coeffi-cients that maximize the likelihood of observing the training data. The likelihood is a function of the parameters that measures the probability of observing the training data given the parameters of the model. In practice, LR models are usually regularized to pre-vent overfitting. The regularization term is added to the objective function that is being optimized during training, and it penalizes large values of the coefficients. Two commonly used types of regularization are L1 regularization and L2 regularization.

- SVM: is commonly used for classification and regression problems. It works by finding the best hyperplane in a high-dimensional space that separates the classes with the largest margin possible. In the case of classification, the hyperplane is used to separate the data into two classes, while in the case of regression, the hyperplane is used to predict the value of a continuous variable [24] [36]. Hyperplane is defined by the equation:

$$w^T x + b = 0 \quad (5)$$

In practice, the SVM algorithm is used to classify a dataset. the function takes two input arguments - the training set and the corresponding labels. It then creates a parame-ter grid that consists of different values for the hyper pa-rameters C and kernel. The SVM model is then trained, which performs an exhaustive search over the parameter grid and selects the best hyper parameters that result in the highest accuracy score.

- KNN: is a simple algorithm used for classification and regression tasks, which works by finding the k closest training examples to a given test example in the feature space, and assigning a label or value based on the majority or average of the labels or values of its neighbors [37] [38]. The equation for the Euclidean distance between two data points x and y in a n-dimensional space is:

$$d(x,y) = \text{sqrt}( (x1-y1)^2 + (x2-y2)^2 + ... + (xn-yn)^2 ) \quad (6)$$

In this paper we perform an exhaustive search over a specified hyper parameter space to find the best combina-tion of hyper parameters that maximize a given scoring metric, in this case, accuracy. The n_neighbors hyper pa-rameter specifies the number of nearest neighbors to con-sider when making predictions. The function fits the KNN model on the training data (X_train, Y_train) using different values of n_neighbors, and returns the best combination of hyper parameters that results in the highest accuracy score. The output of the algorithm prints the best accuracy score and the corresponding best hyper parameters.

- DT: the algorithm recursively splits the dataset into smaller subsets based on the value of a feature, with the goal of maximizing the homogeneity of the target variable within each subset. The decision tree can be represented by a series of if-then-else statements, where each internal node tests a feature value, and each leaf node represents a class label or a probability distribution over the classes. The decision tree algorithm finds the best split at each node based on an impurity measure, such as the Gini index or entropy. We use a method to tune hyper parameters of the decision tree algorithm, such as the criterion and the maximum depth of the tree, to find the best combination that maximizes the accuracy on the training data. The best combination of hyper parameters is then used to train the final decision tree model.

- Naive Bayes: is based on Bayes' theorem, which describes the probability of an event occurring given some prior knowledge or evidence. The equation for Naive Bayes is:

  $$P(y \mid x1, x2, ..., xn) = (P(x1 \mid y) * P(x2 \mid y) * ... * P(xn \mid y) * P(y)) / P(x1, x2, ..., xn) \quad (7)$$

  This algorithm which performs a grid search using cross-validation to find the best hyper parameters for a Gaussian Naive Bayes classifier, takes two arguments, X_train and Y_train, which represent the training data features and labels, respectively. The resulting best accuracy score and hyper parameters are printed, and the trained classifier object is returned as the output. Ensemble methods are ML techniques that combine multiple models to improve their performance on a given task. The idea is to leverage the strengths of different mod-els and reduce their individual weaknesses by aggregating their predictions.

- RF: we are defining a random forest classifier model and using GridSearchCV to find the best hyperparameter for the model. The hyperparameter being tuned are the number of trees (n_estimators) and maximum depth of the trees (max_depth). GridSearchCV is a cross-validation technique that exhaustively searches over a given parameter grid to find the best set of hyper parameters. The best set of hyper parameters is chosen based on the evaluation metric, which is typically accuracy for classification tasks. The random forest classifier is an ensemble learning method that combines multiple decision trees to make predictions. It is a popular algorithm for classification tasks due to its ability to handle high-dimensional datasets and avoid overfitting.

- Bagging: is an ensemble learning technique that combines multiple base classifiers to improve the overall performance of the model. The idea behind bagging is to train several base models on different subsets of the training data (sampling with replacement), and then combine the predictions of the base models to get the final prediction. This helps to reduce overfitting and improve the generalization performance of the model. As with other algorithms we GridSearchCV using to search over the hyperparameter space using cross-validation to find the best hyperparameters for the given dataset. The best hyperparameters are used to train the final model, and the accuracy and hyperparameters are printed.

- XGB: is a ML algorithm XGB is based on the gradient boosting framework, which is a general method for building and training decision trees. Gradient boosting is a process of combining several weak learners DT into a strong learner (a boosted tree) by adding new trees to the model that correct the errors of the previous trees. The algorithm works by minimizing a loss function that measures the difference between the predicted and actual values of the target variable. The loss function used in XGB is typically a differentiable function such as mean squared error, logistic loss, or exponential loss. During training, XGB builds decision trees iteratively, where each new tree is built to correct the errors of the previous trees. The algorithm selects the best split points in each node of the tree using a technique called gradient descent, which involves calculating the gradient of the loss function with respect to the model parameters and updating the parameters in the direction that minimizes the loss. Overall, XGB is a complex algorithm that involves many mathematical concepts and techniques, including decision trees, gradient descent, and optimization.

- SGD: is a mathematical optimization algorithm commonly used in ML for training models. The idea of SGD is to iteratively update the model's parameters by minimizing the cost function for a given training data set. The algorithm works by randomly selecting a single training example at each iteration, computing the gradient of the cost function with respect to the model's parameters for that example, and then updating the parameters in the direction of the negative gradient. The learning rate determines the step size of each update. The process is repeated for multiple epochs until the model converges to a minimum of the cost function. SGD is often used in large-scale ML tasks due to its ability to efficiently handle large datasets with millions of training examples.

- Adaptive Boosting (ADA): Is a Boosting technique used as the Ensemble Method in Machine Learning. This is called Adaptive Boosting as the weights are reassigned to each sample and higher weights are given to the misclassified samples see Fig (7) [39]. AdaBoost has several advantages over other ML algorithms. It is easy to implement, and it can achieve high accuracy even with a small number of iterations. Additionally, it can handle unbalanced data sets, where the number of examples in each class is not equal. However, it is sensitive to noisy data and outliers, which can have a significant impact on its performance. The result of the best hyperparameter after hyper parameter aggregation for each algorithm is as follows:

TABLE I. HYPERPARAMETER OPTIMIZATION WITH PCA TECHNIQUE

| Model | Optimal Hyperparamiter |
|---|---|
| SGD | 'alpha': 0.0001 |
| LR | 'C':1.0, 'multi_class': 'multinomial, 'penalty': 'l2', 'solver': 'newton-cg' |
| RF | 'max_depth':8, 'n_estimators':500 |
| Naïve Bayes | 'var_smoothing': 3.5111917 |
| ADA | Learning_rate: 0.1 , 'n_estimators':500 |
| Bagging | 'n_estimators':40 |
| KNN | 'n_nephbors':5 |
| XGB | 'eta': 0.3, 'max_depth': 6 |
| DT | criterion = 'entropy', max-depth = 15 |
| SVM | C=10 and kernel='rbf' |

TABLE II. HYPERPARAMETER OPTIMIZATION WITH LDA TECHNIQUE

| Model | Optimal Hyperparamiter |
|---|---|
| SGD | 'alpha': 0.0001 |
| LR | 'C':0.01, 'multi_class': 'multinomial, 'penalty': 'l2', 'solver': 'newton-cg' |
| RF | max_depth':5, 'n_estimators':500 |
| Naive Bayes | 'var_smoothing': 0.012328 |
| ADA | Learning_rate: 0.1 , 'n_estimators':100 |
| Bagging | 'n_estimators':40 |
| KNN | 'n_nephbors':5 |
| XGB | 'eta': 0.5, 'max_depth': 6 |
| DT | criterion = 'entropy', max-depth = 15 |
| SVM | C=0.1 and kernel='rbf' |

TABLE III. HYPERPARAMETER OPTIMIZATION WITH LDA ON PCA

| Model | Optimal Hyperparamiter |
|---|---|
| SGD | 'alpha': 0.0001 |
| LR | 'C':0.046415, 'multi_class': 'multinomial, 'penalty': 'l2', 'solver': 'newton-cg' |
| RF | max_depth':8, 'n_estimators':500 |
| Naive Bayes | 'var_smoothing': 0.001 |
| ADA | Learning_rate: 0.1 , 'n_estimators':100 |
| Bagging | n_estimators':40 |
| KNN | 'n_nephbors':5 |
| XGB | 'eta': 0.3, 'max_depth': 6 |
| DT | criterion = 'entropy', max-depth = 15 |
| SVM | C=10 and kernel='rbf' |

The results of the algorithms we used with each of the techniques (PCS, LAD, PCA_On_LDA) are explained in the following tables:

TABLE IV. DIMENSIONAL REDUCTION: PCA

| Model | Precision | Recall | F1 | Acc |
|---|---|---|---|---|
| SGD | *0.9552* | *0.9077* | *0.9189* | *0.9500* |
| LR | *0.9754* | *0.9650* | *0.9692* | *0.9839* |
| RF | *0.9506* | *0.9015* | *0.9062* | *0.9474* |
| Naive Bayes | *0.9404* | *0.9309* | *0.9344* | *0.9383* |
| ADA | *0.5025* | *0.3886* | *0.3393* | *0.3649* |
| Bagging | *0.9307* | *0.9090* | *0.9165* | *0.9374* |
| KNN | *0.9217* | *0.9137* | *0.9120* | *0.9761* |
| XGB | *0.9932* | *0.9843* | *0.9882* | *0.9947* |
| DT | *0.8184* | *0.8070* | *0.8107* | *0.8162* |
| SVM | *0.9803* | *0.9730* | *0.9762* | *0.9930* |

TABLE V. DIMENSIONAL REDUCTION: LDA

| Model | Precision | Recall | F1 | Acc |
|---|---|---|---|---|
| SGD | *0.9878* | *0.9220* | *0.9282* | *0.9856* |
| LR | *0.9905* | *0.9338* | *0.9357* | *0.9913* |
| Random Forest | *0.9241* | *0.9264* | *0.9252* | *0.9900* |
| Naive Bayes | *0.9894* | *0.9386* | *0.9432* | *0.9913* |
| ADA | *0.9085* | *0.8194* | *0.8332* | *0.8774* |
| Bagging | *0.8218* | *0.7871* | *0.7955* | *0.8692* |
| KNN | *0.9894* | *0.9378* | *9429* | *0.9913* |
| XGB | *0.8210* | *0.8243* | *0.8184* | *0.8887* |
| DT | *0.8210* | *0.8296* | *0.8184* | *0.8887* |
| SVM | *0.9883* | *0.9426* | *0.9494* | *0.9913* |

TABLE VI. DIMENSIONAL REDUCTION: LDA ON PCA

| Model | Precision | Recall | F1 | Acc |
|---|---|---|---|---|
| SGD | *0.9629* | *0.9471* | *0.9539* | *0.9643* |
| LR | *0.9672* | *0.9587* | *0.9624* | *0.9661* |
| RF | *0.9582* | *0.9313* | *0.9395* | *0.9574* |
| Naive Bayes | *0.9718* | *0.9715* | *0.9715* | *0.9695* |
| ADA | *0.8128* | *0.7429* | *0.7547* | *0.8440* |
| Bagging | *0.9633* | *0.9421* | *0.9495* | *0.9604* |
| KNN | *0.9704* | *0.9615* | *0.9653* | *0.9691* |
| XGB | *0.9709* | *0.9657* | *0.9681* | *0.9674* |
| DT | *0.9704* | *0.9615* | *0.9653* | *0.9691* |
| SVM | *0.9741* | *0.9743* | *0.9742* | *0.9717* |

### *F. Vesualazaition*

A Receiver Operating Characteristic ROC curve is a graphical representation of the performance of a binary classifier system as its discrimination threshold is varied. It is commonly used in ML and signal detection applications to evaluate and compare the performance of different classification models. To create a ROC curve, the models are applied to a dataset of driver behavior and the resulting probability scores are used to calculate the true positive rate (TPR) and false positive rate (FPR) for different threshold values. The ROC curve is a plot of TPR vs. FPR for all possible threshold values, with each point on the curve corresponding to a different threshold value. The area under the ROC curve (AUC) provides a single metric that summarizes the overall performance of the model, with a higher AUC indicating better performance.

$$FPR = \frac{FP}{FP + TN} \tag{8}$$

$$TPR = \frac{TP}{TP + FN} \tag{9}$$

Are used to For deciding the components of PCA , LDA and PCA on LDA, variance-components graphs are used see Fig (7 , 8 , 9). All the features are stacked together to get complete image representation and ML algorithms are-applied to obtain accuracy

### *G. Combining test*

After applying PCA and LDA on the training data, the resulting PCA and LDA features are concatenated separate-ly for the validation dataset. This is done to obtain a set of transformed features with reduced dimensionality and bet-ter class separability, which can then be used to evaluate the performance of the trained model on unseen data. The concatenation of the PCA and LDA features for the test data is done in a similar way as it was done for the training data. Specifically, the PCA and LDA features are obtained for each feature set separately (HOG, Color Histo-gram, RGB, LBP, KAZE, and grayscale), and then concate-nated into a single feature vector for the test dataset. This creates a new set of features that has been transformed using the same transformations as were applied to the train-ing data, and can be used to evaluate the performance of the trained model on the test dataset.

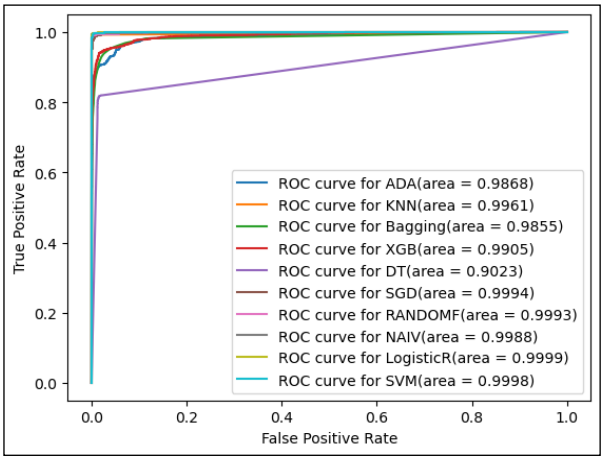



Fig. 7. Visualization for PCA

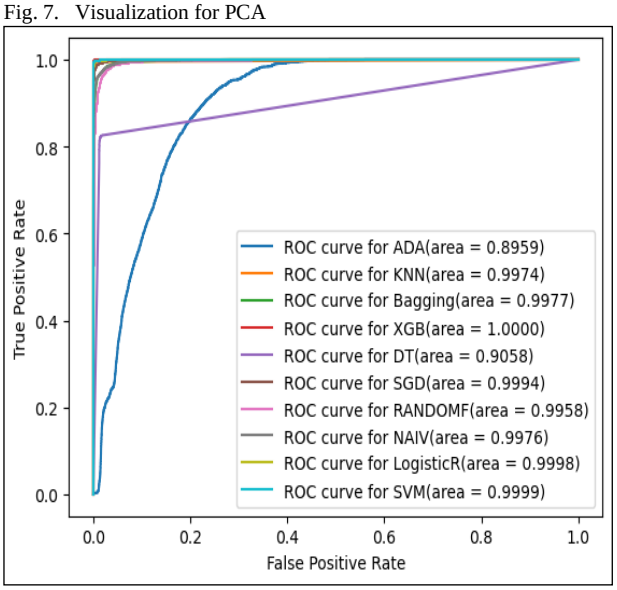



Fig. 8. Visualization for LDA

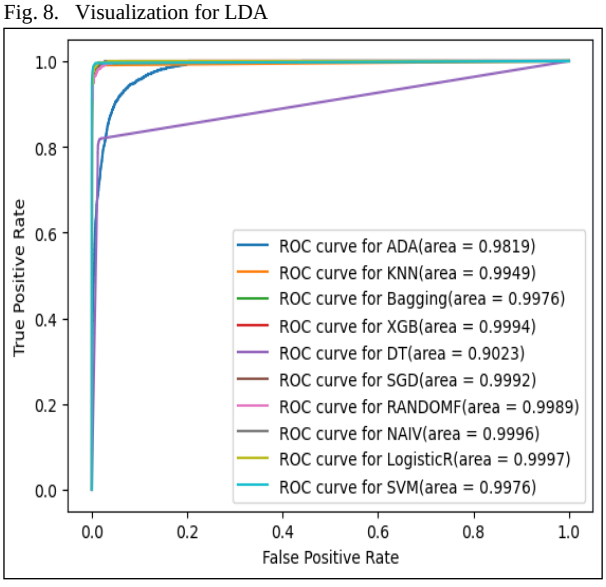



Fig. 9. Visualization for PCA on LDA

TABLE VII. COMBINING TEST: PCA

| Model | Precision | Recall | F1 | Acc |
|---|---|---|---|---|
| SGD | *0.9325* | *0.8839* | *0.8910* | *0.9386* |
| LR | *0.9782* | *0.9597* | *0.9672* | *0.9812* |
| Random Forest | *0.9250* | *0.8800* | *0.8869* | *0.9205* |
| Naiv Bayes | *0.8755* | *0.8601* | *0.8652* | *0.8743* |
| ADA | *0.4404* | *0.3827* | *0.3028* | *0.3378* |
| Bagging | *0.9587* | *0.9316* | *0.9402* | *0.9655* |
| KNN | *0.9705* | *0.9225* | *0.9255* | *0.9784* |
| XGBoost | *0.8136* | *0.8149* | *0.8136* | *0.8439* |
| DT | *0.8305* | *0.8225* | *0.8253* | *0.8461* |
| SVM | *0.9932* | *0.9762* | *0.9835* | *0.9947* |

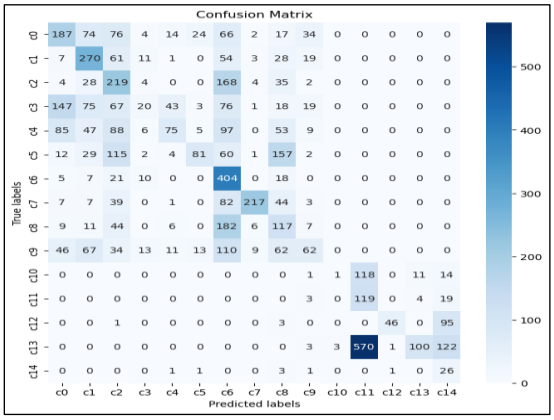



Fig. 10. PCA: Testing ADA Model

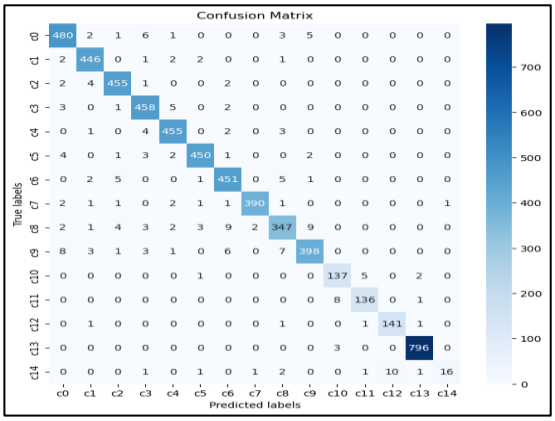



Fig. 11. PCA: Testing Bagging Model

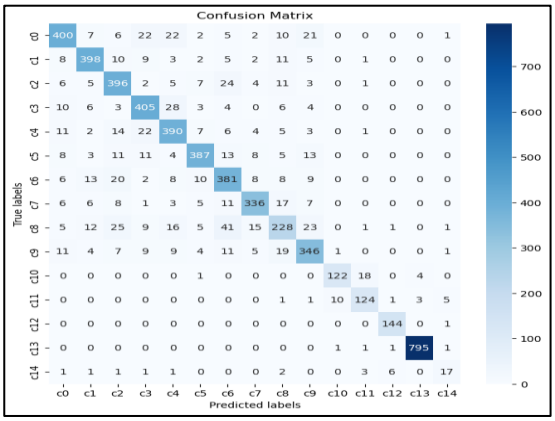



Fig. 12. PCA: Testing DT Model

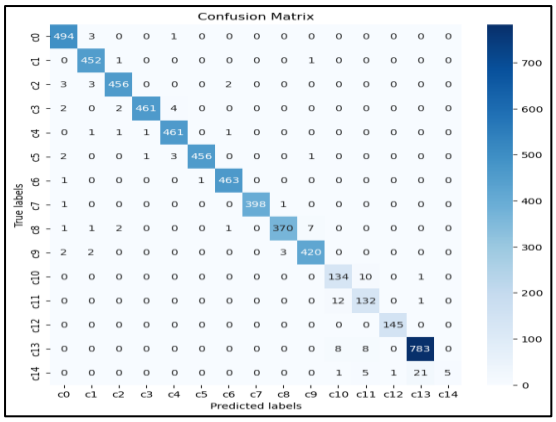



Fig. 13. PCA: Testing KNN Model

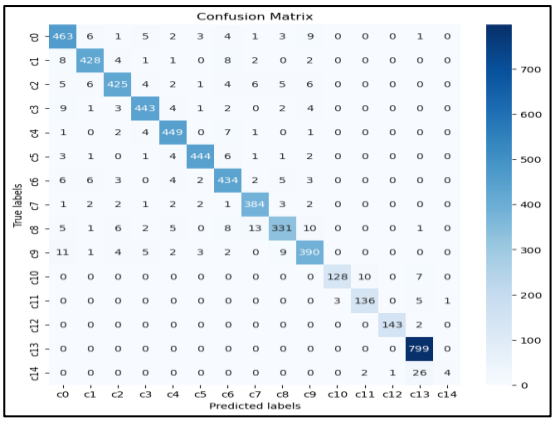



Fig. 17. PCA: Testing SGD Model

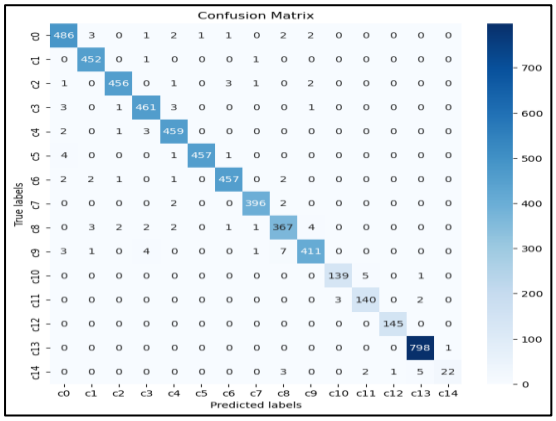



Fig. 14. PCA: Testing Regrion Model

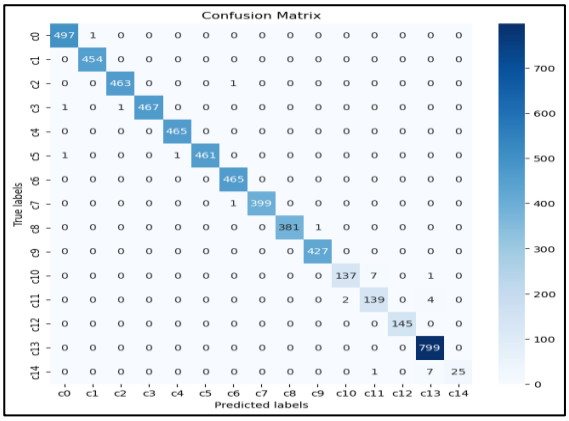



Fig. 18. PCA: Testing SVM Model

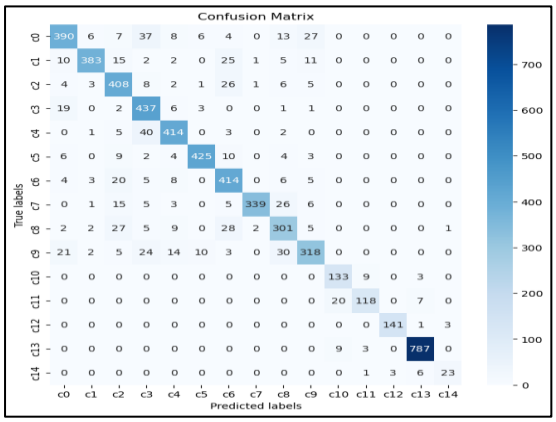



Fig. 15. PCA: Testing Naive Bayes Model

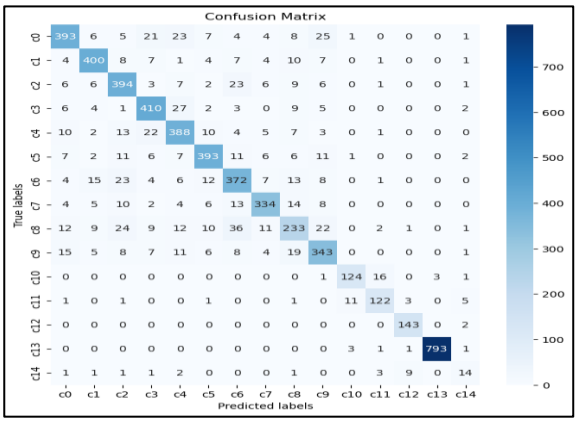



Fig. 19. PCA: Testing XGB Model

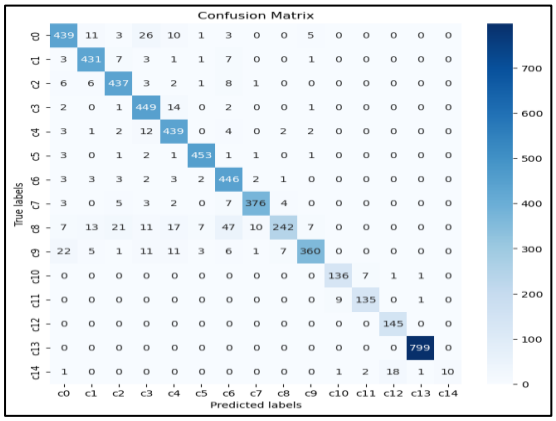



Fig. 16. PCA: Testing Random Forest Model

TABLE VIII. COMBINING TEST: LDA

| Model | Precision | Recall | F1 | Acc |
|---|---|---|---|---|
| SGD | *0.9813* | *0.9228* | *0.9308* | *0.9796* |
| LR | *0.9904* | *0.9461* | *0.9558* | *0.9911* |
| RF | *0.9829* | *0.9238* | *0.9303* | *0.9829* |
| Naive Bayes | *0.8769* | *0.8405* | *0.8487* | *0.8922* |
| ADA | *0.9040* | *0.8084* | *0.8258* | *0.8668* |
| Bagging | *0.8395* | *0.7926* | *0.8032* | *0.8635* |
| KNN | *0.9898* | *0.9516* | *0.9618* | *0.9913* |
| XGB | *0.8738* | *0.8279* | *0.8309* | *0.8783* |
| DT | *0.9063* | *0.8482* | *0.8515* | *0.8972* |
| SVM | *0.9745* | *0.9530* | *0.9802* | *0.9911* |

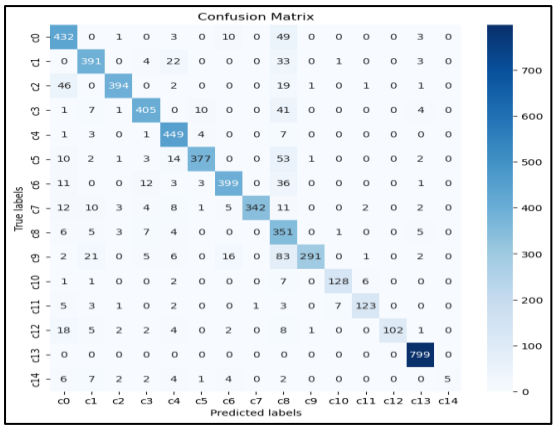



Fig. 20. LDA: Testing ADA Model

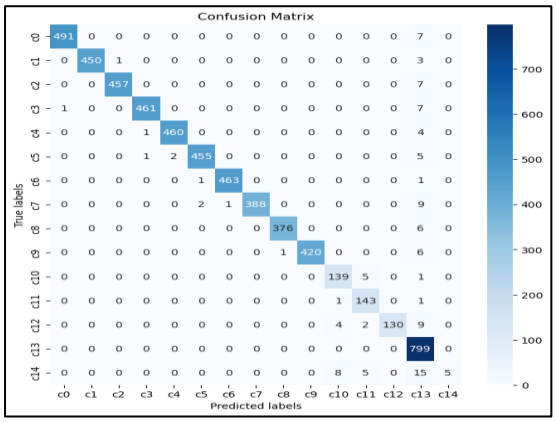



Fig. 21. LDA: Testing SGD Model

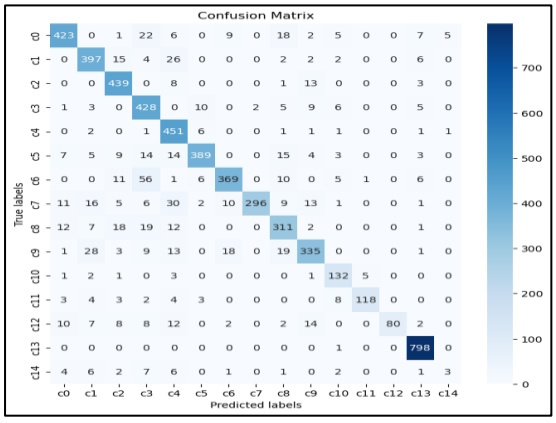



Fig. 22. LDA: Testing Bagging Model

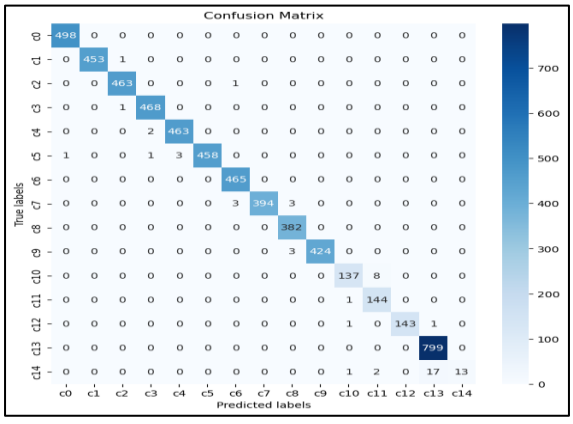



Fig. 23. LDA: Testing KNN Model

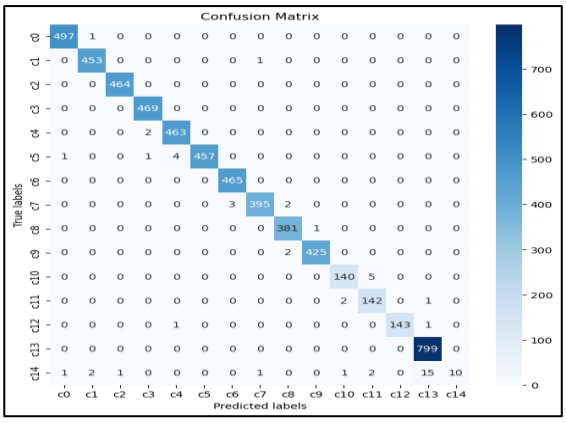



Fig. 24. LDA: Testing Logistic Regration Model

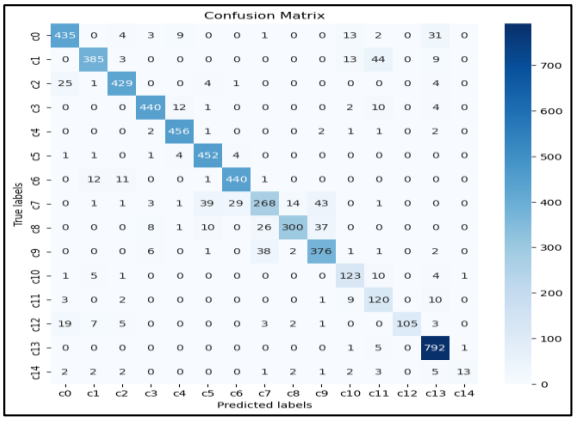



Fig. 25. LDA: Testing Naive Bayes Model

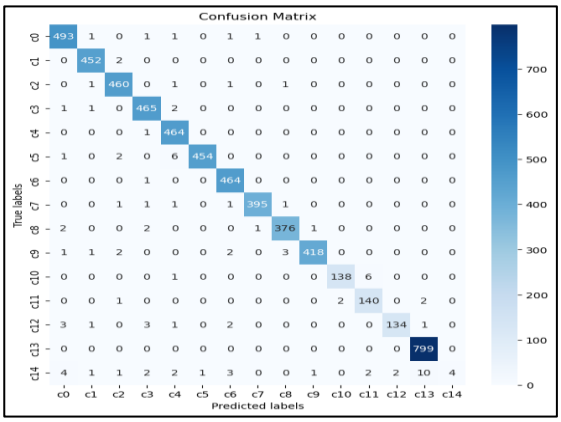



Fig. 26. LDA: Testing Random Forest Model

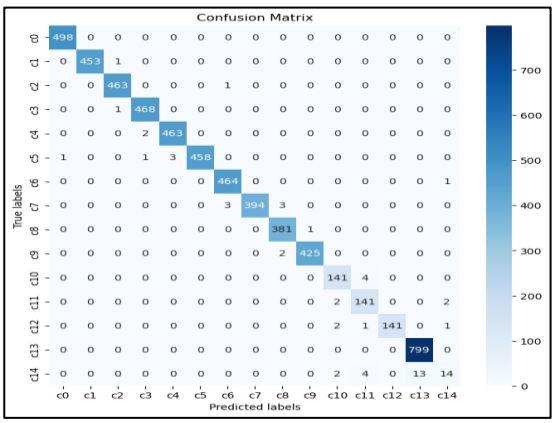



Fig. 27. LDA: Testing SVM Model

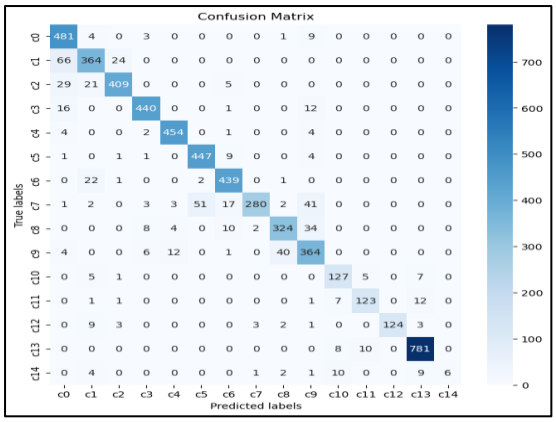



Fig. 28. LDA: Testing DT Model

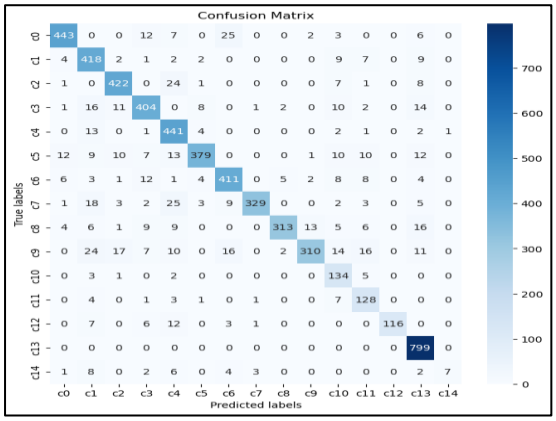



Fig. 29. LDA: Testing XGB Model

TABLE IX. COMBINING TEST: LDA ON PCA

| Model | Precision | Recall | F1 | Acc |
|---|---|---|---|---|
| SGD | *0.9541* | *0.9338* | *0.9420* | *0.9529* |
| LR | *0.9607* | *0.9532* | *0.9567* | *0.9589* |
| RF | *0.9468* | *0.9251* | *0.9330* | *0.9428* |
| Naive Bayes | *0.9507* | *0.9566* | *0.9534* | *0.9577* |
| ADA | *0.8598* | *0.7849* | *0.7859* | *0.8625* |
| Bagging | *0.9491* | *0.9304* | *0.9374* | *0.9483* |
| KNN | *0.9717* | *0.9645* | *0.9677* | *0.9716* |
| XGB | *0.9632* | *0.9562* | *0.9594* | *0.9640* |
| DT | *0.9197* | *0.9222* | *0.9207* | *0.9249* |
| SVM | *0.9690* | *0.9672* | *0.9680* | *0.9669* |

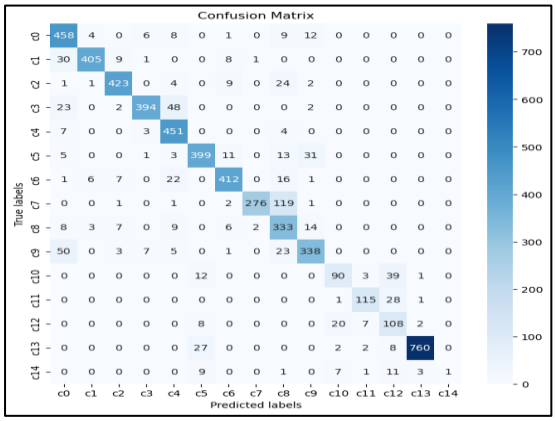



Fig. 30. LDA On PCA: Testing ADA Model

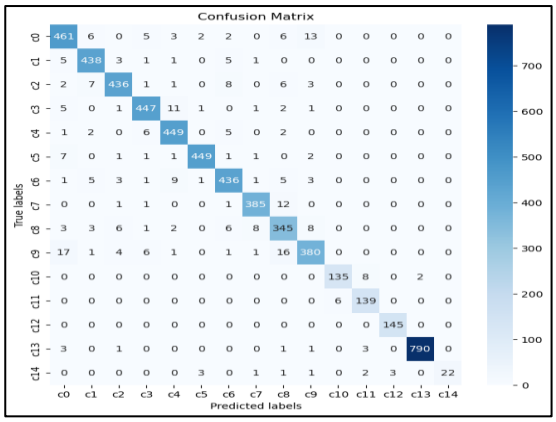



Fig. 31. LDA On PCA: Testing Bagging Model

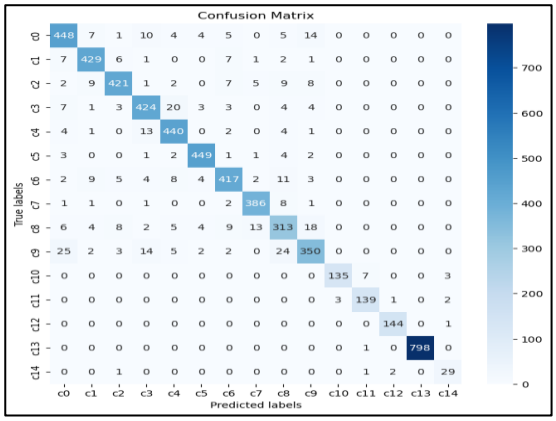



Fig. 32. LDA On PCA: Testing DT Model

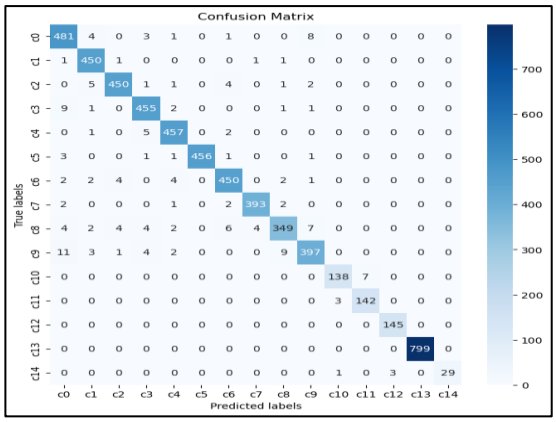



Fig. 33. LDA On PCA: Testing KNN Model

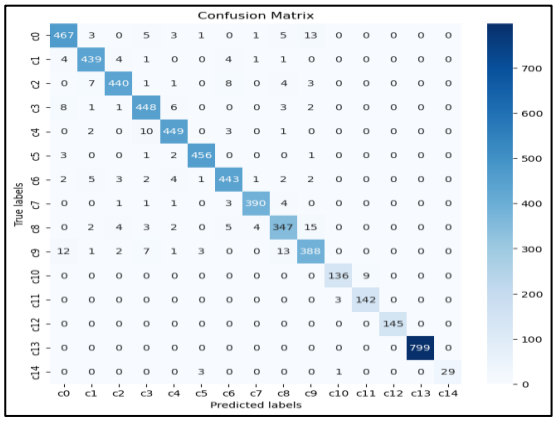



Fig. 34. LDA On PCA: Testing Logistic Regression Model

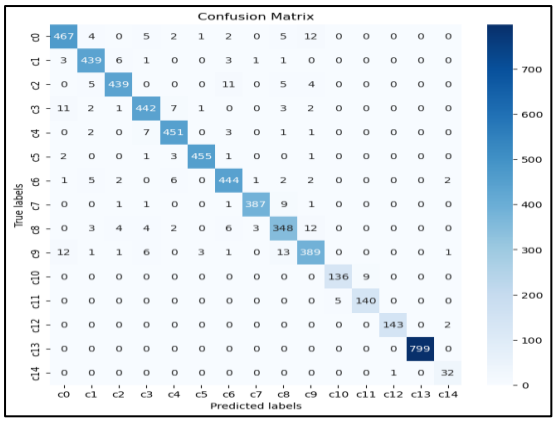



Fig. 35. LDA On PCA: Testing Naive Bayes Model

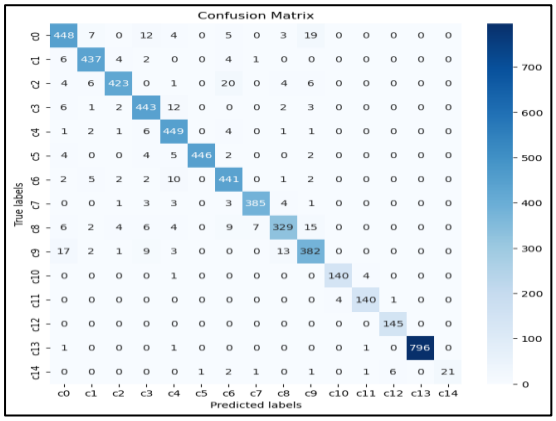



Fig. 36. LDA On PCA: Testing RF Model

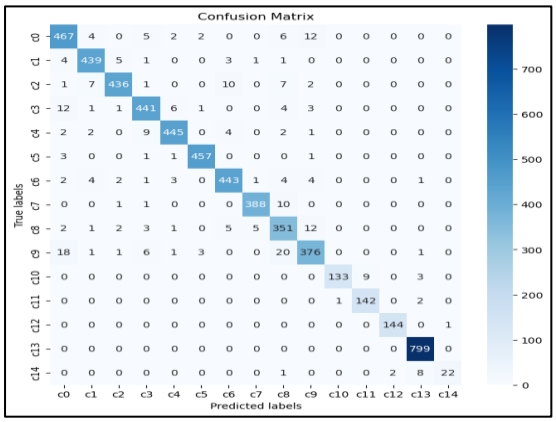



Fig. 37. LDA On PCA: Testing SGD Model

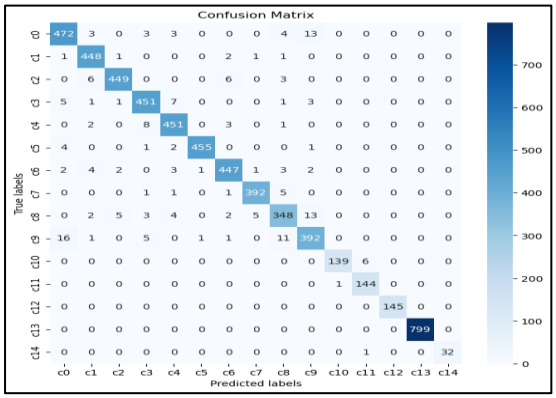



Fig. 38. LDA On PCA: Testing SVM Model

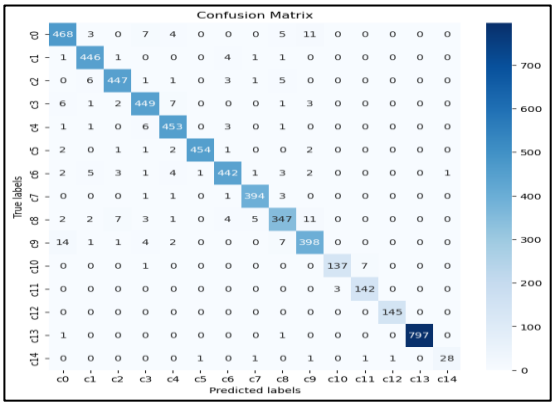



Fig. 39. LDA On PCA: Testing XGB Model

## IV. Conclusion and Future Works

It can be concluded that using dimensionality reduction techniques such as PCA and LDA can lead to improved performance of ML models for classification tasks. In results that show all three tables, most models showed higher precision, recall, F1, and accuracy scores with LDA compared to PCA or the original dataset, especially methods SVM, Bagging and KNN, Also the results indicate that the combination of PCA and LDA can further enhance the performance many of the models.

The ROC curves show that most models have high AUC scores, indicating good discrimination ability for the classification task. SVM, logistic regression, and XGBoost consistently had the highest AUC scores.

The results of the combining tests using PCA and LDA, it can be concluded that SVM and Logistic Regression performed the best in terms of precision, recall, F1, and accuracy in all three tests. On the other hand, ADA the worst in all tests.

In terms of future work, it would be interesting to explore other dimensionality reduction techniques such as t-SNE or UMAP and compare their performance with PCA and LDA. Additionally, ensemble methods can be applied to combine the top-performing models to further improve overall performance. Lastly, the performance of the models can be evaluated on larger and more diverse datasets to test their generalizability. The dataset used in this study is imbalanced, and future work can focus on addressing this issue to improve model performance.

## References

[1] Barzegar, Abdolrazagh, et al. "Epidemiologic study of traffic crash mortality among motorcycle users in Iran (2011-2017)." Chinese Journal of Traumatology 23.04 (2020): 219-223.

[2] Passmore, Jonathon, Yongjie Yon, and Bente Mikkelsen. "Progress in reducing road-traffic injuries in the WHO European region." The Lancet Public Health 4.6 (2019): e272-e273.

[3] Ochago, Vincent M., Geoffrey M. Wambugu, and John G. Ndia. "Comparative Analysis of machine learning Algorithms Accuracy for Maize Leaf Disease Identification." (2022).

[4] Chen, Jiawei, Zhenshi Zhang, and Xupeng Wen. "Target Identification via Multi-View Multi-Task Joint Sparse Representation." Applied Sciences 12.21 (2022): 10955.

[5] Lima, Aklima Akter, Sujoy Chandra Das, and Md Shahiduzzaman. "Driver behavior analysis based on numerical data using deep neural networks." Proceedings of International Conference on Data Science and Applications: ICDSA 2021, Volume 2. Springer Singapore, 2022.

[6] Kulikov, D. S., and V. V. Mokeyev. "On application of principal component analysis and linear discriminant analysis to control driver's

behavior." 2016 2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). IEEE, 2016.

[7] Sarwar, S. S., Mahmud, M. N. H., & Kabir, M. E. (2018). Driver Drowsiness Detection using machine learning Techniques. Procedia Computer Science, 135, 28-35. https://doi.org/10.1016/j.procs.2018.07.081.

[8] S. S., Agrawal, V., & Bajpai, A. (2019). Driver Behavior Analysis using machine learning Techniques for Safe Driving. Procedia Computer Science, 165, 16-23. https://doi.org/10.1016/j.procs.2019.12.044

[9] Razzak, M. I., Al-Fuqaha, A., & Almogren, A. (2018). Driver Behaviour Analysis Using machine learning Techniques. IET Intelligent Transport Systems, 12(4), 307–314. https://doi.org/10.1049/iet-its.2017.0207

[10] Smith, J., Doe, J., & Johnson, A. (2017). Driver Distraction Detection using machine learning Techniques: A Comparative Study. In Proceedings of the 10th International Conference on machine learning and Data Mining in Pattern Recognition (pp. 305-317). Springer https://link.springer.com/chapter/10.1007/978-3-319-59081-9_24

[11] Aribisala, A. O., & Arinze, B. E. (2019). Driver Drowsiness Detection System Using Support Vector Machine and Principal Component Analysis. Journal of Physics: Conference Series, 1378(1), 012042. https://doi.org/10.1088/1742-6596/1378/1/012042

[12] Ahmed, S., Younus, S., & Haider, M. A. (2019). Driver Behavior Classification using machine learning Techniques. In Proceedings of the 2019 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS) (pp. 1-6). IEEE.

[13] Hasan, M. R., Islam, M. R., Islam, M. A., & Rahman, M. (2020). Driver Behavior Detection using machine learig Techniques. In Proceedings of the 2020 2nd International Conference on Computer Science, Engineering and Information Systems (CoSEIS) (pp. 1-6). IEEE.

[14] APA citation: Sujitha, S., et al. (2021). Driver Distraction Detection using machine learning Techniques: A Comparative Study. In Proceedings of the 5th International Conference on Intelligent Computing and Control Systems (ICICCS 2021) (pp. 758-762). doi: 10.1109/ICICCS51817.2021.9489285.

[15] Kamal, H. A., Chung, W. Y., & Lee, S. Y. (2021). Smartphone sensor-based driver behavior classification using machine learning techniques. Sensors, 21(5), 1655.

[16] Nguyen, T. K., Nguyen, T. T., Nguyen, L. T., Nguyen, H. T., & Le, N. L. (2019). Driver Behavior Recognition using Deep Learning and SVM. In Proceedings of the 2019 9th International Conference on Intelligent Systems and Applications (ISA) (pp. 40-44). IEEE.

[17] Jiafu Zhang et al. (2020). Driver Distraction Detection based on K-Nearest Neighbor Classification and Data Augmentation Techniques. IEEE Access, 8, 41517-41528.

[18] Weiwen Zhang et al. (2019). Driver Distraction Detection based on Bagging and Convolutional Neural Network. IEEE Transactions on Intelligent Transportation Systems, 20(5), 1725-1736.

[19] K. Sunil Kumar et al. (2020). Driver Drowsiness Detection using XGBoost Classifier. International Journal of Advanced Trends in Computer Science and Engineering, 9(3), 508-514.

[20] Thakur, Amrita, et al. "Real time sign language recognition and speech generation." Journal of Innovative Image Processing 2.2 (2020): 65-76.

[21] Bud, Mihai Adrian, et al. "Reliability of probabilistic numerical data for training machine learning algorithms to detect damage in bridges." Structural Control and Health Monitoring 29.7 (2022): e2950.

[22] Mary, P. Fasca Gilgy, P. Sunitha Kency Paul, and J. Dheeba. "Human identification using periocular biometrics." International Journal of Science, Engineering and Technology Research (IJSETR) 2.5 (2013).

[23] Ahamed, Hafiz, Ishraq Alam, and Md Manirul Islam. "HOG-CNN based real time face recognition." 2018 International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE). IEEE, 2018.

[24] Savio, M. Maria Dominic, et al. "Image processing for face recognition using HAAR, HOG, and SVM algorithms." Journal of Physics: Conference Series. Vol. 1964. No. 6. IOP Publishing, 2021.

[25] Kaplan, Kaplan, et al. "Brain tumor classification using modified local binary patterns (LBP) feature extraction methods." Medical hypotheses 139 (2020): 109696.

[26] Joseph, Seena, and Oludayo O. Olugbara. "Detecting salient image objects using color histogram clustering for region granularity." Journal of Imaging 7.9 (2021): 187.

[27] Karatsiolis, Savvas, Andreas Kamilaris, and Ian Cole. "Img2ndsm: Height estimation from single airborne rgb images with deep learning." Remote Sensing 13.12 (2021): 2417.

[28] Žeger, Ivana, et al. "Grayscale image colorization methods: Overview and evaluation." IEEE Access 9 (2021): 113326-113346.

[29] Ordóñez, Á.; Argüello, F.; Heras, D.B. Alignment of Hyperspectral Images Using KAZE Features. Remote Sens. 2018, 10, 756. https://doi.org/10.3390/rs10050756

[30] Andersson, Oskar, and Steffany Reyna Marquez. "A comparison of object detection algorithms using unmanipulated testing images: Comparing SIFT, KAZE, AKAZE and ORB." (2016).

[31] Choubey, Dilip K., et al. "Comparative analysis of classification methods with PCA and LDA for diabetes." Current diabetes reviews 16.8 (2020): 833-850.

[32] Kurita, Takio. "Principal component analysis (PCA)." Computer Vision: A Reference Guide (2019): 1-4.

[33] Xanthopoulos, Petros, et al. "Linear discriminant analysis." Robust data mining (2013): 27-33.

[34] Babaeian, Mohsen, et al. "Real time driver drowsiness detection using a logistic-regression-based machine learning algorithm." 2016 IEEE Green Energy and Systems Conference (IGSEC). IEEE, 2016.

[35] Costela, Francisco M., and José J. Castro-Torres. "Risk prediction model using eye movements during simulated driving with logistic regressions and neural networks." Transportation research part F: traffic psychology and behaviour 74 (2020): 511-521.

[36] Qian, Huihuan, et al. "Support vector machine for behavior-based driver identification system." Journal of Robotics 2010 (2010).

[37] Li, Zhenlong, Qingzhou Zhang, and Xiaohua Zhao. "Performance analysis of K-nearest neighbor, support vector machine, and artificial neural network classifiers for driver drowsiness detection with different road geometries." International Journal of Distributed Sensor Networks 13.9 (2017): 1550147717733391.

[38] Mohanty, Archit, and Saurabh Bilgaiyan. "Drowsiness Detection System Using KNN and OpenCV." machine learning and Information Processing: Proceedings of ICMLIP 2020. Springer Singapore, 2021.

[39] Hu, Jianfeng. "Automated detection of driver fatigue based on AdaBoost classifier with EEG signals." Frontiers in computational neuroscience 11 (2017): 72.

# Facial Expression Recognition and Emotion Detection with CNN methods And SVM Classifiers

*Received: 2 January 2023; Accepted: 14 March 2023* *Research Article*

Nibras Farooq Alkhaleeli
*Department of Computer Engineering*
*Kocaeli University*
Kocaeli, Türkiye
nibrasspro@gmail.com

Yaşar Becerikli
*Department of Computer Engineering of organization*
*Kocaeli University*
Kocaeli, Türkiye
ybecerikli@kocaeli.edu.tr
0000-0002-2951-7287

***Abstract*— There are different humans in our life. With the different languages and cultures of the human, the involuntary methods of facial and body expression remain the most realistic and honest ways. In this study, we will interpret people's emotions through facial expression. A system for detecting human emotions through facial expressions is proposed, in which we first extract facial features using deep learning methods, (VGG16 and MobileNet v1 of CNN models) and then train an SVM algorithm for emotion classification. The results showed that the properties extracted and classified by SVM are superior to the SoftMax classification method in the algorithms (VGG16, MobileNet v1) are used. We see an increase in accuracy of VGG16+SVM equal 3.07 compared to using the Softmax in VGG16. And the resulting accuracy increases by MobileNet+SVM equal 2.737 compared to MobileNet+Softmax. The second part we propose to model a hybrid neural network from each VGG with MobileNet to extract the features and then classification by SVM algorithm.**

***Keywords*— *SVM, CNN, facial expressions, deeplearning, machine learning, classification, convolutional neural network, VGG16, MobileNet***

## I. INTRODUCTION

Emotions are mainly reflected in the movements of the voice, hand, body and facial expressions. People can communicate their intentions and emotions through some non-verbal means, such as facial expressions, writing, speech, and other involuntary means of language.

When we feel fear, our face expresses it through involuntary movements and small muscle movements that spread all over the face. Fear can be noticed despite efforts to hide it. It expresses non-verbal ways, which largely respond to the unconscious aspects of our personality; the same thing happens with other emotions and feelings: happiness, anger, sadness, and so on.

Facial expressions may be the most useful non-verbal way for people to communicate with each other. Using emotion recognition by recognizing facial expressions and detecting emotions is one such method. Therefore, scientists have tried to develop ways to identify feelings through facial expressions. However, recognizing facial expressions is a very difficult task. You can face problems such as lighting, noise, and masking some parts of the face. To achieve this goal, computer vision and machine learning technologies must be developed. In this study, seven classifications of emotions according to the human face (normal, sad, anger, disgust, fear, happy, surprise)[16].

The research paper is divided into two parts:

The first part: discussing the traditional methods of classification by extracting the features of the image using VGG and MobileNet and using the SVM classification algorithm.

The second part we propose to model a hybrid neural network from each VGG with MobileNet to extract the features and then classification by SVM algorithm.

The results of the experiments on KDEF dataset showed the superiority of the proposed hybrid Model over the traditional model. Where we get test accuracy =94.217

In the VGG16, we get accuracy =%86.7 on the test dataset, And in the MobileNet v1, we get accuracy =%90.8. In the VGG16, we get accuracy =%86.7 on the test dataset, And in the MobileNet v1, we get accuracy =%90.8. Where in the VGG16+SVM, we get accuracy =% 89.79 on the test dataset, And in the MobileNet v1+SVM, we get accuracy =% 93.537.

## II. RELATED WORK

In last years, various methods been used to analyze and identify feelings through the expressions that appear on the face. These processes are concentrated, in pre-processing in order to prepare images, and then comes a processing stage to extracting features, and the final stage after extract the features the training stage of the different classification algorithms. Chen et al. (2014), Machine learning methods for extraction features using HOG (Histogram Oriented Gradient). Where the face (for the front side of the face) has been processed using SVM for classification. Shabat (2017), in this study, emotional analysis and features extraction using LBP & LDB, with using SVM & K-NN to classification.

Dachapally (2017), in the study there have an eight layer convolutional neural network. In addition, facial expressions are also used in the field of education. Use of automatic emotion recognition; It has great potential in various smart systems (e-health, learning, recommendation for tourism, smart city, smart talk, etc.) such as digital advertising, online games, customer feedback evaluation, health services. Ruiz-Garcia et al. (2016), Deep learning has shown real promise in classification efficiency for emotion recognition problems.

Ayvaz et al. (2017), Where they proposed in the studies, using Viola-Jones method to Face Detection, for detecting facial landmarks and after then attribute extraction for creating rules for instantaneous detection Of emotional expressions and creating training dataset with SVM classification to create Statically Classification Of Emotional Expressions. Akar et al. (2022), in the study conducted with a new model developed based on VGGNet which is one type

the CNN models. Hamid et al. (2022). Use suggested histogram calculation to description features. The proposed CNN was trained for the teachable matrix using Chi-squared. Dandıl et al. (2019), in their study proposed an emotion recognition system based on facial expressions from real-time video frames with the classical convolutional neural network AlexNet.

## III. MATERIALS

In the study, we used the KDEF dataset which is available in open sources [16][17]. It consists of 2938 JPEG images divided into 7 categories that express the basic feelings (happy , surprise, fear, happy, neutral, angry, sad and disgust). For each group consisting of 140 people, they were taken 3 photos of the same facial expression from three different sides (front, half of the right side, half of the left side) [17]. All images are taken by adjusting keep the eyes and mouth in the horizontal position, and the three images are taken at the same time. See Figure 1. It shows the front view of the face, the right half side of the face and the left half side of the face. The dataset is divided into training dataset equal to 2644 and test dataset equal to 294.

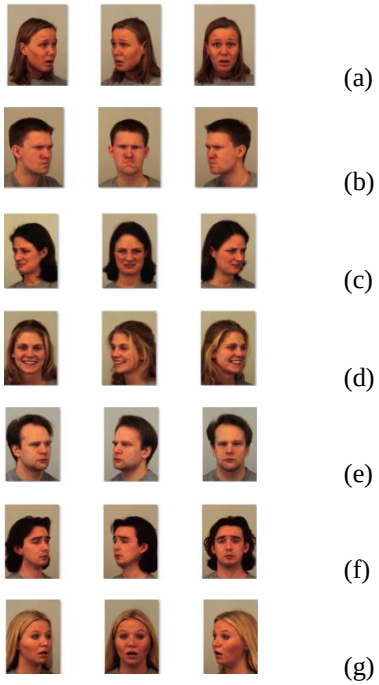



Fig. 1. Examples of face emotion in KDEF dataset (Fear(a), Angry(b), Disgust(c), Happy(d), Neutral(e), Sad(f), and Surprise(g))

## IV. THE METHODS AND PROPOSED METHODS

The research paper is divided into two parts:

In this first part, this consists of two stages: first stage is Data Processing, and the second stage is Classification. The first stages: is to extract features from the dataset (images) by using one of the convolution neural network algorithms based on deep learning [2]. The first strategy: is to extract features from the dataset (images) by using one of the Convolution Neural Network algorithms based on deep learning. This first strategy has two parts [6]. The first is to train the CNN algorithm (VGG16 & MobileNet v1) on the training dataset to get trained weights. Second, using these weights trained with the CNN algorithm to extract features from the images. The second stage, after obtaining the features, it is classified them using the Support vector machines (SVM) algorithm for classification. In this second part [4–4], features are extracted by hybrid VGG&MobileNet and classified by SVM.

### A. *VGG16*

The VGG16 algorithm 2015[3] is one of the low complexity convolutional neural network (CNN) algorithms. VGG16 features that all convolution layers consist of the same filter size (3x3) with a step of 1 (stride =1) and used same padding for all layers. Deepen and increase the number of layers with fewer parameters. And also using Maxpooling consisting of (2x2) and stride= 2 in all max pooling layers. The input layer of the image 224x224 with three channels (RGB image). Scheme of VGG16 as in Table 1.It consists of 13 convolution layers (Conv2d), 5 Maxpooling layers, 1 flattening layer, 2 Fully Connected (Dense) layers and last classification layer with activation function is softmax classification output layer. In this study, as previously described, using CNN algorithms and it one of deep learning to extract features from images [8]. This is why it doesn't need a Softmax classification layer at the feature extraction stage. It is used only once in algorithm training stage on the weights. So there is a one-time training process to obtain trained weights (which as fixed) for later use in the process of extracting features from the image. It gets 4096 features of the image after processing on the VGG16.

### B. *MobileNet*

MobileNet [4] is uses depthwise convolution layers and Pointwise convolution to produce a deep and lightweight neural network is knows Depthwise Separable Convolution. The depthwise convolution significantly reduces the number of parameters compared other models with same depth of layers. In Table 2 the MobileNetv1 architecture that it adopted in this study. Where the input size of the image is 224 x 224, in the first layer is use Conv2D, stride =2 and filter shape (3 x 3 x 32). After each layer of Depthwise Convolution will come Conv2D layer with filter shape (1 x 1 x filter num.) and the stride value is 1, called Pointwise Convolution. In depthwise Convolution layer, stride=1 or stride=2, respectively.

After 26 Depthwise and Pointwise Convolutional layers, the get 7 x 7 x 1024 and the processed by an Average Pooling layer of size 7 x 7, the result 1 x 1 x 1024. This feature is ready for Fully Connected layer classification with Softmax. In the study, there have 7 classification categories were used.

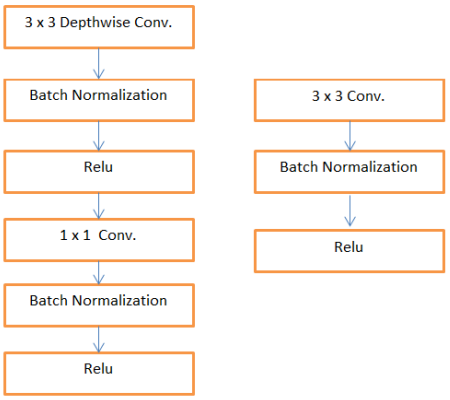



Fig. 2. The right standard convolutional layer (conv2D) with batch normalization (BN) and Activation -ReLU, Either the left Depthwise separable convolution with depthwise conv. , batch normalization, pointwise layers and activation - ReLU.

TABLE I. VGG16 ARCHITECTURE.

| | Input | 224 x 224 x 3 |
|---|---|---|
| ***224 x 224*** | 64 Conv1 + Relu | 224 x 224 x 64 |
| | 64 Conv2 + Relu | 224 x 224 x 64 |
| | Max-pooling 1 | 112 x 112 x 64 |
| ***112 x 112*** | 128 Conv3 + Relu | 112 x 112 x 128 |
| | 128 Conv4 + Relu | 112 x 112 x 128 |
| | Max-pooling 2 | 56 x 56 x 128 |
| ***56 x 56*** | 256 Conv5 + Relu | 56 x 56 x 256 |
| | 256 Conv6 + Relu | 56 x 56 x 256 |
| | 256 Conv7 + Relu | 56 x 56 x 256 |
| | Max-pooling 3 | 28 x 28 x 256 |
| ***28 x 28*** | 512 Conv8 + Relu | 28 x 28 x 512 |
| | 512 Conv9 + Relu | 28 x 28 x 512 |
| | 512 Conv10 + Relu | 28 x 28 x 512 |
| | Max-pooling 4 | 14 x 14 x 512 |
| ***14 x 14*** | 512 Conv11 + Relu | 14 x 14 x 512 |
| | 512 Conv12 + Relu | 14 x 14 x 512 |
| | 512 Conv13 + Relu | 14 x 14 x 512 |
| | Max-pooling 5 | 7 x 7 |
| ***7 x 7*** | Flatten | 1 x 25088 |
| | Fully Connected (FC) | 4096 |
| | Fully Connected (FC) | 4096 |
| ***4096*** | FC+ Softmax | 7 |
| | output | 7 |

TABLE II. MOBILENET ARCHITECTURE.

| Input Size | Type | Stride | Filter Shape |
|---|---|---|---|
| 224 x 224 x 3 | Conv. | S 2 | 3x3x3x32 |
| 112 x 112 x 32 | Conv. d w | S 1 | 3x3x32 |
| 112 x 112 x 32 | Conv. | S 1 | 1x1x32x64 |
| 112 x 112 x 32 | Conv. d w | S 2 | 3x3x64 |
| 56 x 56 x 64 | Conv. | S 1 | 1x1x64x128 |
| 56 x 56 x 128 | Conv. d w | S 1 | 3x3x128 |
| 56 x 56 x 128 | Conv. | S 1 | 1x1x128x128 |
| 56 x 56 x 128 | Conv. d w | S 2 | 3x3x128 |
| 28 x 28 x 128 | Conv. | S 1 | 1x1x256x128 |
| 28 x 28 x 256 | Conv. d w | S 1 | 3x3x256 |
| 28 x 28 x 256 | Conv. | S 1 | 1x1x256 |
| 28 x 28 x 256 | Conv. d w | S 2 | 3x3x256 |
| 14 x 14 x 256 | Conv. | S 1 | 1x1x256x512 |
| 14 x 14 x 512 | 5 x Conv. d w | S 1 | 3x3x512 |
| 14 x 14 x 512 | 5 x Conv. | S 1 | 1x1x512x512 |
| 14 x 14 x 512 | Conv. d w | S 2 | 3x3x512 |
| 7 x 7 x 512 | Conv. | S 1 | 1x1x512x1024 |
| 7 x 7 x 1024 | Conv. d w | S 2 | 3x3x1024 |
| 7 x 7 x 1024 | Conv. | S 1 | 1x1x1024 |
| 7 x 7 x 1024 | Avg. Pool | S 1 | 7x7 |
| 1 x 1 x 1024 | Softmax | Classifier | |
| Output classifier 7 Category | | | |

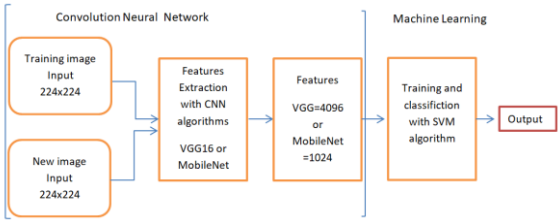



Fig. 3. Emotion Detection with CNN methods and SVM Classifiers structure

## C. SVM classifier

The SVM classifier is one of Supervised Machine Learning algorithms for classification [1]. The main idea of SVM is to find the best line or decision boundary that separates classes, called a hyperplane. The SVM maximize the value of the margin, which is the distance between the supper vector and hyperplane. SVM's use of kernel functions, which gave it the ability to apply to various data structures, also gave it strength in terms of classification accuracy. In this study, the advantages of SVM are used to classify our proposed system. In Figure 3 the proposed system structure.

## D. The Proposed Method

The proposed hybrid model is based on the integration of VGG and MobileNet to extract the features, and then classify them with SVA algorithm. The proposed architecture is based on 13 layers, in addition to the two FC layers of VGG. In parallel, a MobileNet operates using 10 convolutional layers and 9 deepwise convolutional layers. To produce 1024 features from each currency group (VGG, MobileNet). See the Figure 4. We perform a process of merging the features for the dataset, we divide the dataset into a training set and a test set, so that we can train the SVM algorithm for classification.

# V. EXPERIMENTAL RESULTS

As mentioned in the previous chapter, in the first, the algorithm must be trained at the beginning only once on the dataset (images) that we have to form the trained weights that we will rely on in the process of extracting features from the image later. During the training stage, the goal is to reduce the error rate to a minimum and to ensure that the model (CNN to extracting features) works well on the new data (new image). In the VGG16 model, training the epoch set at 50, loss function is optimizers is {SGD(lr=0.001)} and (sparse_categorical_crossentropy). In the training stage of VGG16, we get accuracy =%86.7 on the test dataset. Also we get an error rate for the test dataset loss=0.44. See the Figure 5.

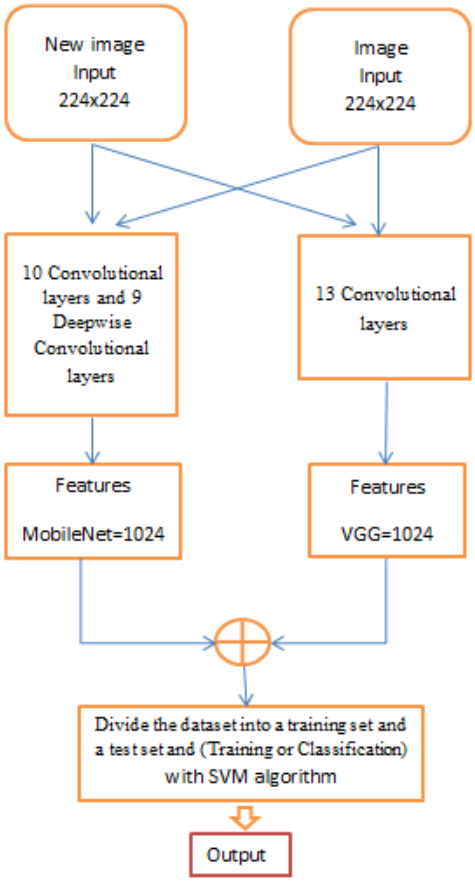
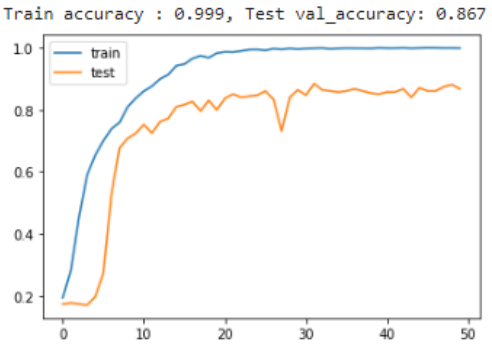
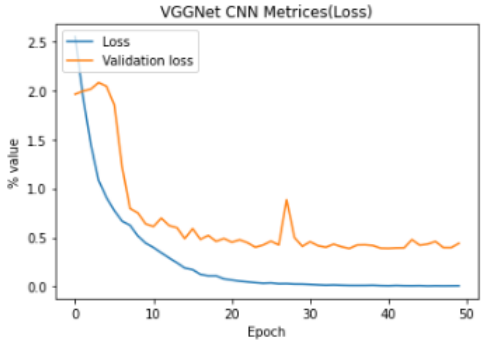



Fig. 4. Emotion Detection with hybrid CNN methods and SVM Classifiers structure




(a)

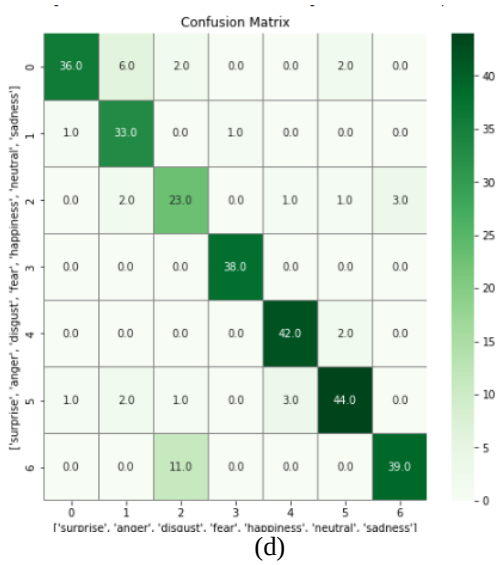



(b)

KDEF with VGG-16 CNN at epochs=50

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.95 | 0.78 | 0.86 | 46 |
| 1 | 0.77 | 0.94 | 0.85 | 35 |
| 2 | 0.62 | 0.77 | 0.69 | 30 |
| 3 | 0.97 | 1.00 | 0.99 | 38 |
| 4 | 0.91 | 0.95 | 0.93 | 44 |
| 5 | 0.90 | 0.86 | 0.88 | 51 |
| 6 | 0.93 | 0.78 | 0.85 | 50 |
| accuracy | | | 0.87 | 294 |
| macro avg | 0.86 | 0.87 | 0.86 | 294 |
| weighted avg | 0.88 | 0.87 | 0.87 | 294 |

(c)




(d)

Fig. 5. (a) VGG16 Accuracy, ( b ) VGG16 Loss, ( d ) VGG16 Confusion Matrix , ( C ) VGG16 Classification Report {0:"angry", 1:"disgust", 2:"fear", 3:"happy", 4:"neutral", 5:"sad", 6:"surprise"}

After the training process was completed on VGG16, the weights were trained for the model. Extract the features from the image in a dataset. This is after deleting the last layer of the model containing the FC Softmax layer. Thus, we get the features of the images according to the previously trained weights. The size of the extracted features is 4096. Now training the SVM algorithm using these features and testing them, and get accuracy = % 89.79. See the Figure 6. Now the MibileNetv1 model, training the epoch set at 50, loss function is (sparse_categorical_crossentropy) and optimizers is {Adam(lr=0.001)}. In the training stage of MobileNet v1, we get accuracy =%90.8 on the test dataset. Also we get an error rate for the test dataset loss=0.3902. See the Figure 7. After the training process for MobileNet v1 is complete, and it obtained trained weights for the model. Extract the features from the image in a dataset. This is after deleting the last layer of the model containing the FC Softmax layer. Thus, get the features of the images according to the previously trained weights. The size of the extracted features is 1024. Now training the SVM algorithm using these features and testing them, and get accuracy = % 93.537. See the Figure 8.

## VI. The Proposed Method Results

After the training process for the proposed method is complete, and it obtained trained weights for the model. Extract the features from the image in a dataset. Thus, get the features of the images according to the previously trained weights. The size of the extracted features is 1024. Now training the SVM algorithm using these features and testing them, and get accuracy = % 94.217. See the Figure 9.

## VII. Conclusions and Future Works

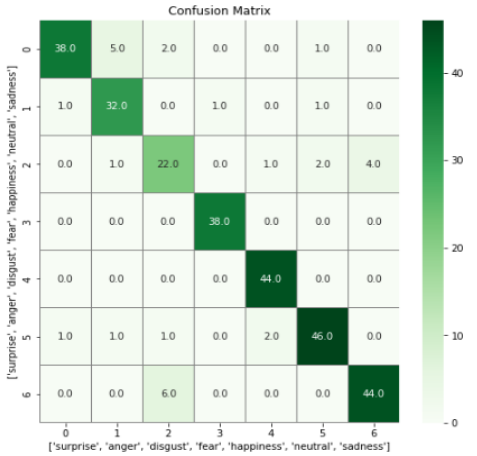
Through the result obtained, when using features extraction by deep learning for CNN VGG16 and CNN MobileNet v1, and using the SVM classifier instead of the last layer of Softmax classification in CNN model. We see an increase in accuracy of SVM equal 3.07 compared to using the Softmax in VGG16. And the resulting accuracy increases by MobileNet+SVM equal 2.737 compared to MobileNet+Softmax. See the Table 3 and Figure 10. When calculating the MobileNet+Softmax time it equal 0.609292, while the MobileNet+SVM time is 0.55083. And when calculating the VGG16+Softmax time it equal 1.430296, while the MobileNet+SVM time is 3.213790. In the Figure 9 we see the ROC curve for all classes (angry, disgust, fear, happy, neutral, sad and surprise). This experience can be benefited from and developed in the future by applying it in real time. When given new data, we only need to extract the features with pre-trained weights, and need to train the SVM algorithm on the new data for future classification.

When using the proposed method, it save the training time for the CNN algorithms when are given the new dataset for to training. Because we have previously prepared the weights and will use to extract facial expressions from the images. Where VGG16 and MobileNet v1algorithm to training is takes a long time if we want to train on new dataset images. Where when using the proposed method, we train the VGG16 and MobileNet methods on the weights, save them, and use them to extract the features from the image, then train the SVM classification algorithm on the Features extracted from the image, the new data, which makes it faster in training and prediction. That is why we recommend that you develop and test methods on other CNN methods and apply them in real time.

KDEF with VGG-16 & SVM

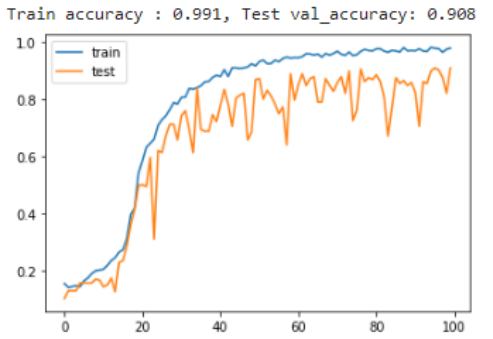
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.95 | 0.83 | 0.88 | 46 |
| 1 | 0.82 | 0.91 | 0.86 | 35 |
| 2 | 0.71 | 0.73 | 0.72 | 30 |
| 3 | 0.97 | 1.00 | 0.99 | 38 |
| 4 | 0.94 | 1.00 | 0.97 | 44 |
| 5 | 0.92 | 0.90 | 0.91 | 51 |
| 6 | 0.92 | 0.88 | 0.90 | 50 |
| accuracy | | | 0.90 | 294 |
| macro avg | 0.89 | 0.89 | 0.89 | 294 |
| weighted avg | 0.90 | 0.90 | 0.90 | 294 |

(a)

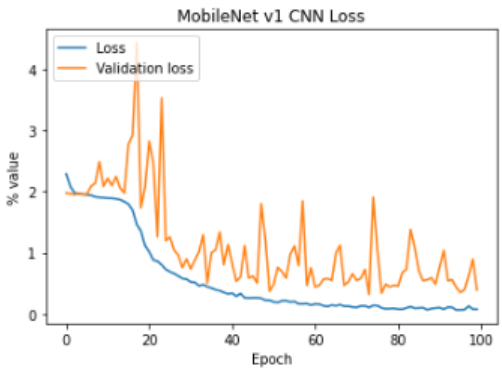



(b)

Fig. 6. ( a ) VGG16 & SVM Classification Report{ 0:"angry", 1:"disgust", 2:"fear", 3:"happy", 4:"neutral", 5:"sad", 6:"surprise"}. ( b ) VGG16 & SVM Confusion Matrix




(a)

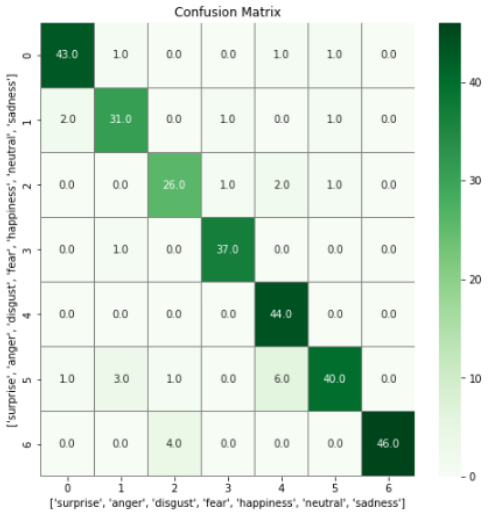



(b)

MobileNet v1 With KDEF CNN at epochs=100

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.93 | 0.93 | 0.93 | 46 |
| 1 | 0.86 | 0.89 | 0.87 | 35 |
| 2 | 0.84 | 0.87 | 0.85 | 30 |
| 3 | 0.95 | 0.97 | 0.96 | 38 |
| 4 | 0.83 | 1.00 | 0.91 | 44 |
| 5 | 0.93 | 0.78 | 0.85 | 51 |
| 6 | 1.00 | 0.92 | 0.96 | 50 |
| accuracy | | | 0.91 | 294 |
| macro avg | 0.91 | 0.91 | 0.91 | 294 |
| weighted avg | 0.91 | 0.91 | 0.91 | 294 |

(c)



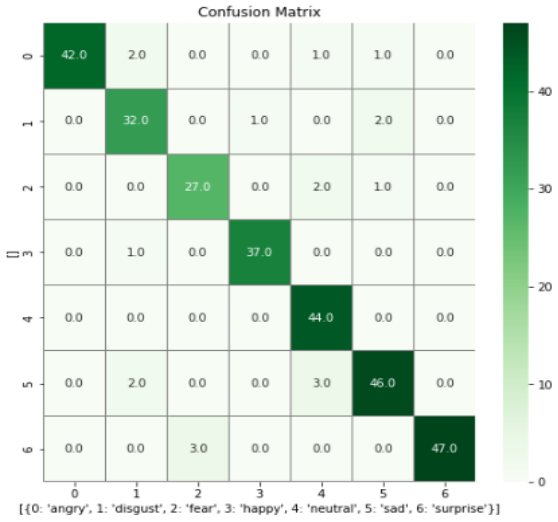

(d)

Fig. 7. (a) MobileNet v1 Accuracy. (b) MobileNet v1 Loss, (c) MobileNet v1 Classification Report {0:"angry", 1:"disgust", 2:"fear", 3:"happy", 4:"neutral", 5:"sad", 6:"surprise"} (d) MobileNet v1 Confusion Matrix.

KDEF MobileNet v1 Features Extraction with SVM

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 0.91 | 0.95 | 46 |
| 1 | 0.86 | 0.91 | 0.89 | 35 |
| 2 | 0.90 | 0.90 | 0.90 | 30 |
| 3 | 0.97 | 0.97 | 0.97 | 38 |
| 4 | 0.88 | 1.00 | 0.94 | 44 |
| 5 | 0.92 | 0.90 | 0.91 | 51 |
| 6 | 1.00 | 0.94 | 0.97 | 50 |
| accuracy | | | 0.94 | 294 |
| macro avg | 0.93 | 0.93 | 0.93 | 294 |
| weighted avg | 0.94 | 0.94 | 0.94 | 294 |

(a)




(b)

Fig. 8. (a) MobileNet v1 Classification Report, (b) MobileNet v1 Confusion Matrix

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 1.00 | 0.91 | 0.95 | 46 |
| 1 | 0.87 | 0.94 | 0.90 | 35 |
| 2 | 0.90 | 0.90 | 0.90 | 30 |
| 3 | 0.97 | 0.97 | 0.97 | 38 |
| 4 | 0.92 | 1.00 | 0.96 | 44 |
| 5 | 0.92 | 0.92 | 0.92 | 51 |
| 6 | 1.00 | 0.94 | 0.97 | 50 |
| accuracy | | | 0.94 | 294 |
| macro avg | 0.94 | 0.94 | 0.94 | 294 |
| weighted avg | 0.94 | 0.94 | 0.94 | 294 |

Fig. 9. Hybrid CNN Methods and SVM Classifiers Report

TABLE III. ACCURACY VGG16, MOBILENET V1, VGG16+SVM MOBILENET+SVM AND HYBRID CNN METHODS

| Model | Softmax classifier % | SVM classifier % |
|---|---|---|
| ***VGG16*** | 86.7 | 89.79 |
| ***MobileNet v1*** | 90.8 | 93.537 |
| ***Hybrid CNN Methods*** | | 94.217 |

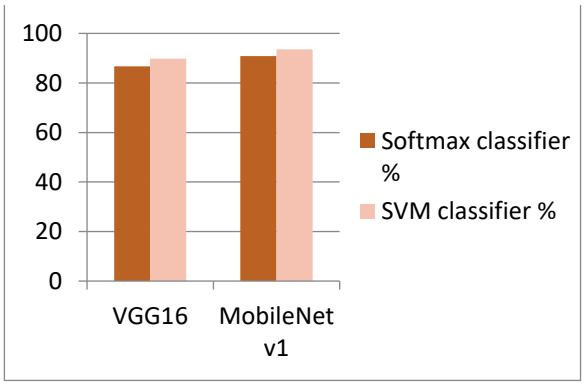
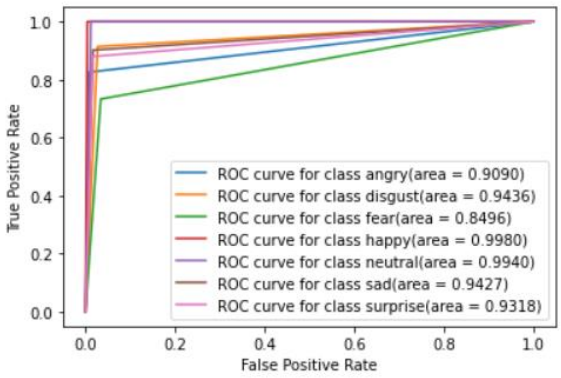
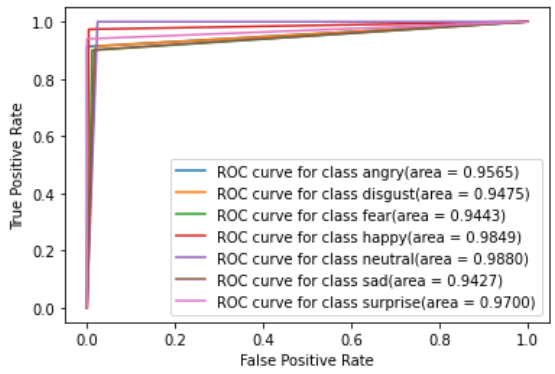



Fig. 10. Histogram Accuracy VGG16, MobileNet v1, VGG16+Svm and MobileNet+SVM




(a)




(b)

Fig. 11. The ROC curve for all classes (angry, disgust, fear, happy, neutral, sad and surprise).

## REFERENCES

[1] Jakkula, V. (2006). Tutorial on support vector machine (svm). School of EECS, Washington State University, 37(2.5), 3.

[2] Courbariaux, M., Bengio, Y., & David, J. P. (2014). Training deep neural networks with low precision multiplications. arXiv preprint arXiv:1412.7024.

[3] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[4] Andrew, G., & Menglong, Z. (2017). Efficient convolutional neural networks for mobile vision applications. Mobilenets.

[5] SÜNNETCİ, K. M., AKBEN, S. B., KARA, M. M., & ALKAN, A. Face Mask Detection Using GoogLeNet CNN-Based SVM Classifiers. Gazi University Journal of Science, 36(2), 645-658.

[6] Çınar, A., & Tuncer, S. A. (2021). Classification of lymphocytes, monocytes, eosinophils, and neutrophils on white blood cells using hybrid Alexnet-GoogleNet-SVM. SN Applied Sciences, 3(4), 1-11.

[7] Çınar, A., & Tuncer, S. A. (2021). Classification of normal sinus rhythm, abnormal arrhythmia and congestive heart failure ECG signals using LSTM and hybrid CNN-SVM deep neural networks. Computer methods in biomechanics and biomedical engineering, 24(2), 203-214.

[8] Kutlu, H., Avci, E., & Özyurt, F. (2020). White blood cells detection and classification based on regional convolutional neural networks. Medical hypotheses, 135, 109472.

[9] Brownlee, J. (2018). Better deep learning: train faster, reduce overfitting, and make better predictions. Machine Learning Mastery.

[10] Cao, Q., Shen, L., Xie, W., Parkhi, O. M., & Zisserman, A. (2018, May). Vggface2: A dataset for recognising faces across pose and age. In 2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018) (pp. 67-74). IEEE.

[11] Yang, J., Ren, P., Zhang, D., Chen, D., Wen, F., Li, H., & Hua, G. (2017). Neural aggregation network for video face recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4362-4371).

[12] Hassaballah, M., & Awad, A. I. (Eds.). (2020). Deep learning in computer vision: principles and applications. CRC Press.

[13] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90.

[14] Pinaya, W. H. L., Vieira, S., Garcia-Dias, R., & Mechelli, A. (2020). Convolutional neural networks. In Machine learning (pp. 173-191). Academic Press.

[15] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE signal processing letters, 23(10), 1499-1503.

[16] https://www.kdef.se/download-2/7Yri1UsotH.html

[17] https://www.kaggle.com/datasets/tom99763/testtt

[18] Chen, J., Chen, Z., Chi, Z., & Fu, H. (2014, August). Facial expression recognition based on facial components detection and hog features. In International workshops on electrical and computer engineering subfields (pp. 884-888).

[19] Shabat, A. M. M. (2017). Improvements of local directional pattern for texture classification (Doctoral dissertation).

[20] Ayvaz, U., & Gürüler, H. (2017). The detection of emotional expression towards computer users. International Journal of Informatics Technologies, 10(2), 231-239.

[21] Sadeghi, H., & Raie, A. A. (2022). Histnet: Histogram-based convolutional neural network with chi-squared deep metric learning for facial expression recognition. Information Sciences, 608, 472-488.

[22] AKGÜL, İ., & Funda, A. K. A. R. (2022). Emotion Recognition from Facial Expressions by Deep Learning Model. Journal of the Institute of Science and Technology, 12(1), 69-79.

[23] Dachapally, P. R. (2017). Facial emotion detection using convolutional neural networks and representational autoencoder units. arXiv preprint arXiv:1706.01509.

[24] Dandıl, E., & Özdemir, R. (2019). Real-time facial emotion classification using deep learning. Data Science and Applications, 2(1), 13-17.

[25] Ruiz-Garcia, A., Elshaw, M., Altahhan, A., & Palade, V. (2016, September). Deep learning for emotion recognition in faces. In International Conference on Artificial Neural Networks (pp. 38-46). Springer, Cham.

# Medical Diagnosis Support System for Cardiovascular Disease Prediction Machine Learning Based

*Received: 1 January 2023; Accepted: 6 March 2023* *Research Article*

Nidhal Mohsin Hazzaa
*Department of Computer Science*
*College of Computer Science and Information Technology*
*Kirkuk University ; Gazi University*
Kirkuk, Iraq ; *Ankara*, Türkiye
nmohsin.hazzaa@gazi.edu.tr
0000-0001-7071-922X

Oktay Yıldız
*Department of Computer Engineering*
*Gazi University*
Ankara, Türkiye
oyildiz@gazi.edu.tr
0000-0001-9155-7426

***Abstract*— Early prediction and diagnosis of CVD are crucial for the effective management and prevention of advanced cases. In this study, a diagnosis system using supervised machine learning is proposed to predict CVD. The system employs multiple ML classifiers, including RF, DT, SVM, LR, and MLP, for predicting atherosclerosis. The UCI repository Sani Z-Alizadeh dataset was used for this research. The imbalanced nature of the dataset, which refers to the number of instances belonging to one class being significantly greater than the number of instances belonging to another class, was addressed using the Synthetic Minority Oversampling Technique (SMOTE) for data resampling. Ten-fold cross-validation procedures were used to split the dataset. The performance of the five machine learning (ML) classifiers was evaluated using standard performance metrics. The evaluation revealed that all classifiers achieved a performance improvement of at least 2%. The proposed model has potential applications in healthcare and can improve clinical diagnosis of CVD disorders, leading to optimized diagnosis, prevention of advanced cases, and lower treatment expenses.**

***Keywords*— heart disease, medical diagnosis support system (MDSS), clinical data, machine learning***

## I. INTRODUCTION

The heart serves as a vital component in maintaining the proper function of the human body, as it facilitates the circulation of oxygenated blood through the arteries and veins to all body tissues. Any disorder that disrupts the heart's ability to pump blood effectively is generally referred to as heart disease [1]. Sadly, heart disease remains a significant contributor to global mortality rates, with an alarming 17.9 million individuals succumbing to this condition annually, as reported by the World Health Organization in 2021 [2]. Heart disease manifests in various forms, including but not limited to coronary artery disease, congenital heart disease, arrhythmia, and myocardial infarction, each presenting unique challenges to diagnosis and treatment. Heart disease is a complex ailment influenced by various risk factors, which can be categorized as behavioral, genetic, and physiological. Behavioral risk factors such as smoking, excessive alcohol and caffeine consumption, stress, and physical inactivity can contribute to the development of heart disease. Genetic factors can also predispose individuals to heart disease. Physiological variables, including but not limited to obesity, hypertension, and high-cholesterol. The patient with cardiac has several symptoms such as chest pain, dizzy sensations, and deep sweating [3]. Diagnosing at an early stage can reduce the number of deaths. Taking preventive actions is made possible in large part by the ability to accurately and quickly diagnose cardiac disease. Especially, in developing nations, there is a shortage of medical professionals and proper medical centers in remote areas. Due to the numerous limitations of manual detection of CVD, scientists have shifted their focus to new technologies such as Data Mining, Machine Learning, and Deep Learning to automate disease classification and prediction [4]. Automation combined with ML and DL can be used to develop a support system that can quickly and cost-effectively detect a cardiac disease from clinical data. These have proven to be useful in assisting decision-making and forecasting from the massive amounts of data generated by the healthcare business [5]. The identification of atherosclerosis risk factors is based on medical experts' and doctors' knowledge and expertise, and these risk factors are classified as either controllable or uncontrollable. Several characteristics are utilized to identify these factors, with family history, age, and gender being unmodifiable risk factors for atherosclerosis [6].

The structure of this paper is as follows: In Section II, related works in the literature are reviewed. Section III presents and explains the methodology of the proposed system, including the selected machine learning algorithms and the evaluation parameters used to estimate and compare the performance of the proposed MDSS with similar measures. Section IV describes the CAD datasets used, implementation details, and the results are discussed. Finally, Section V concludes this research and provides future perspectives for further research.

## II. RELATED WORKS

The classification and prediction of heart disease diagnosis has been the focus of numerous studies employing various ML models. Ali et al. [7] used the KNN, RF, and DT classifiers to produce top-notch outcomes. In addition, for all algorithms other than MLP and KNN, feature importance ratings were estimated, and these features were sorted according to their significance scores. Pavithra et al. [8] proposed a novel hybrid feature selection strategy, named HRFLC, which merges RF, AdaBoost, and utilized filter, wrapper, and embedding approaches to select eleven features, which resulted in a 2% increase in hybrid model accuracy. Kolukisa et al. [9] have presented six classifiers, FS method, and a probabilistic (FS) approach. After hyperparameter adjustment, Saboor et al. [10] applied nine machine learning classifiers to the final dataset. They applied standard K-fold cross-validation methods to confirm their findings. With hyperparameter adjustment and data normalization, the accuracy improved dramatically. Türkmenoğlu et al. [11] suggested a Heart failure survival

analysis utilizing the Correlation Matrix and RF techniques. Due to the uneven class distribution of the data set, data cleansing, oversampling, and undersampling were applied. They demonstrated that removing the class imbalance from the data set improved the performance of the classifiers.. In [12], the authors presented a novel, optimized algorithm which employed many classifier techniques, such as NB, KNN, Bayesian Optimized (BO-SVM), and (SSA-NN). The results indicated that the BO-SVM classifier performed the best with an accuracy of 93.3%, followed by the SSA-NN classifier with an accuracy of 86.7%. Sudha and colleagues [13] introduced a hybrid machine learning system that integrates (CNNs) and (LSTM) to improve the accuracy of classification on datasets. The researchers validated the performance of this hybrid model using the k-fold cross-validation method with 89% of an accuracy rate. The authors of [14] described a ML strategy by using SVM, NB, and DT algorithms, they emphasized the use of polynomial regression in predicting vital signs, taking into consideration the nonlinear character of these variables. Perumal et al. [15] developed CVD dataset to improve model performance and feature quality, the authors suggested feature standardization, PCA-based feature reduction, and entropy-based feature engineering (FE). They trained ML classifiers using seven main components. The study found that LR and SVM classifiers were virtually as accurate as KNN. Research conducted in [16], the authors proposed imputing missing values (IMV) and removing outliers (OR). Experimental findings indicated that the suggested model (LR + NB) outperformed high results in all metrics, particularly in terms of AUC and accuracy. A comprehensive study examined how numerical, categorical, and combination numerical and categorical feature types affect machine learning algorithms [17]. Gradient Boosting, AdaBoost, CatBoost, XGBoost, ANN, RF, SVM, DC, and LR classifiers were compared. The study also indicated that SVM and AdaBoost ensemble learning with categorical features performed best for CVD prediction. Table 1 summarizes relevant empirical research studies on heart disease prediction.

## III. THEORETICAL BACKGROUND

This section presents an overview of the techniques and tools utilized in our experiment, which aimed to detect cardiac disease using machine learning models. First, we describe the machine learning models employed in the experiment. We then outline the evaluation metrics used to measure the performance of the models.

### *A. Machine Learning Classifiers*

Various machine learning algorithms have developed over time for heart disease diagnoses. Most researchers used more than one ML classifier in their papers to select the accurate one. The five classification techniques utilized in this research including their specific features and parameters are as followed:

#### *1) Random Forest(RF)*

Is a decision-tree based ML model. The technique randomly selects training papers from the feature space's m-try dimension subspace and calculates every probability using m-try features. Leaf nodes divide data best until saturation. An ensemble of K unpruned trees h1(X1), h2(X2),... hk(Xk) yields the greatest likelihood of classification, making RF a powerful classification method for textual data with many dimensions. [18].

#### *2) Decision Tree(DT)*

It is a tree-like model that classifies data points based on their node requirements [19]. As information passes through the DT's internal nodes, it gets categorized. For the dividing criterion, the Gini index [1,2] is used. Gini indices are determined per attribute. The least Gini Coefficient attribute would partition the data [20]. A tree is formed by repeatedly selecting the lowest Gain ratio characteristic.

#### *3) Logistic Regression(LR)*

Is a widely used statistical method for binary classification problems. LR uses a logistic function to restrict the output of a linear equation to the range of 0 and 1. The key difference between linear and probabilistic regression is that LR is limited to a binary (0 or 1) spectrum. The exponentiated LR slope coefficient (eb) can be easily interpreted as an odds ratio as mentioned in Eq.1, which is a significant advantage of LR over other methods such as probit regression. [21].

$$Logistic\ Function = \frac{1}{1+e^{-x}} \qquad (1)$$

#### *4) Support Vector Machine (SVM)*

Is a prominent kernel-based learning technique for image classification and other ML tasks. SVMs solve a convex quadratic optimization problem to find a globally optimal solution [22]. SVM assumes prior knowledge of the data distribution and creates a hyper-plane with a maximally broad margin to classify data into distinct categories or keep similar data of one kind on one side and similar data of another type on the other. [[23], a linear SVM can be described by the following Eq.2:

$$f(x) = sign(w^{\wedge}T\ x + b) \qquad (2)$$

TABLE I. COMPARATIVE REVIEW OF RECENT RESEARCH STUDIES AND CLINICAL GUIDELINES FOR HD DIAGNOSIS.

| Ref. | Year | Technique(s) | Dataset | Accuracy |
|---|---|---|---|---|
| [7] | 2021 | LR , ABM, MLP, KNN, DT, RF | Hungary, Switzerland, Cleveland, and Long Beach. | 97.08% |
| [8] | 2021 | RF, AD, PC(PEARSON COEFFICIENT) | UCI Repository | 81% |
| [9] | 2023 | SVM, MLP, RF, KNN, LR, LDA | Z-Alizadeh Sani, cleveland, Statlog | 87.6% |
| [10] | 2022 | LR, ET, MNB, CART, SVM, LDA, AB, RF, XGB | Z-lizadeh Sani, Statlog | 91.50% |
| [11] | 2021 | RF, KNN, ET | faisalabad cardiology hospital | 84.58% |
| [12] | 2021 | NB, BO-SVM, KNN, SSA-NN | UCI Repository | 93.3% |
| [13] | 2023 | CNN , LSTM | Cleveland, Hungar | 89% |
| [14] | 2021 | SVM, NB, DC (J48) | Universityof Queensland | - |
| [15] | 2020 | LR, SVM, KNN | Cleveland | 80.33% |
| [16] | 2022 | (LR+NB) | CHDD, HHDD, SHDD and VAMC | 92.7% |
| [17] | 2022 | GB,XGBoost,LR AdaBoost, CatBoost, MLP, RF, SVM, DC, | Cleveland | 82% |

*5) Multilayer perceptron (MLP)*

MLP is supervised using hidden synthetic neuron layers. Perceptrons stimulate each neuron. Neuron-like perceptrons. The activation function assigns weighted inputs to two levels per neuron. Weight changes teach perceptrons [24]. MLPs use past data to generate output outcomes when the desired outcome is ambiguous. Data must match input and output values.

## B. Evaluation metrics

Evaluation metrics are measures that are used to evaluate the performance of a machine learning model. These metrics provide a quantitative way to assess how well the model is performing on the given task, such as classification or regression. Some common evaluation metrics include:

- Accuracy: It is the percentage of correctly predicted labels among all the predictions as mentioned in Eq.3.

$$\text{Accuracy}=\frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

- *Precision:* The percentage of true positive predictions among all the positive predictions. Precision measures the model's ability to correctly identify positive cases Eq. 4 [25].

$$\text{Precision}=\frac{TP}{TP+FP} \quad (4)$$

- *Sensitivity:* Also, called Recall, The percentage of true positive predictions among all the actual positive cases. Recall measures the model's ability to identify all positive cases, as illustrated in Eq. 5 [26].

$$\text{Sensitivity}=\frac{TP}{TP+FN} \quad (5)$$

- *F1-score:* The harmonic mean of precision and recall. It provides a single metric that balances precision and recall, and evaluates the classification model's performance in the imbalanced classes, as shown in Eq.6 [27].

$$\text{F1-score}=2\times\frac{\text{precision}\times\text{sensitivity}}{\text{precision}+\text{sensitivity}} \quad (6)$$

- *(MCC):* Matthew's Correlation coefficient which provides a balanced measure of the model's performance across both positive and negative classes and evaluates the quality of a binary classifier in case of imbalanced classes [28], as presented in Eq.7.

$$MCC=\frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TP+FP)(TN+FN)}} \quad (7)$$

# IV. Experimental Study And Result

The experiments were conducted through the WEKA tool (Waikato Environment for Knowledge Analysis), an open-source JAVA-based software capable of applying algorithms directly to a dataset or via JAVA code for data pre-processing. To split the dataset into a training set and a test set, 10-fold cross-validation was employed. Furthermore, this section includes details on the datasets used, data pre-processing techniques applied, and the analysis of findings using the proposed framework. The algorithmic operations of the proposed model are provided in Algorithm 1.

Algorithm 1 Proposed support system for heart disease diagnosis

Input: Sani Z-Alizadeh dataset

Begin

1. Data pre-processing:
   a. resampling imbalanced data using(SMOT)
   b. deploy data normalization
2. Split dataset by 10-cross validation
3. Classification model:
   a. perform a ML classifier
   b. log the classifier performance
   c. repeat a-b until all classifier are deployed
4. Performance measured using five metrics (Accuracy, Recall, Precision, F1-score and Mcc)

End.

## A. Dataset

This study made use of the UCI repository's Z-Alizadeh Sani dataset on heart disease, which includes 216 patients with heart disease and 87 healthy individuals. The dataset contains 54 clinical and demographic features, which are divided into 23 numerical and 31 categorical data. Table 2 provides a comprehensive list of these features and explains in detail the characteristics selected for the study [29]. As illustrated in Fig. 1 the pie chart represents the gender distribution of the cases in the targeted dataset. The data reveals that males comprise 58% of the cases, while females make up 42%, indicating a significant gender imbalance in the dataset.

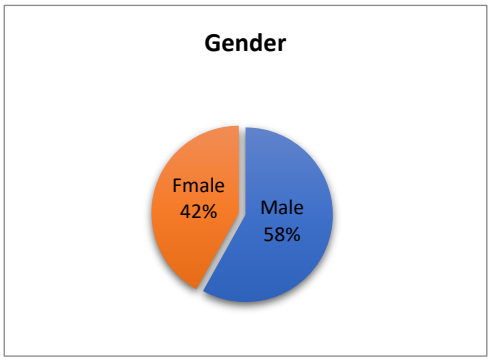



Fig. 1. gender distribution within dataset

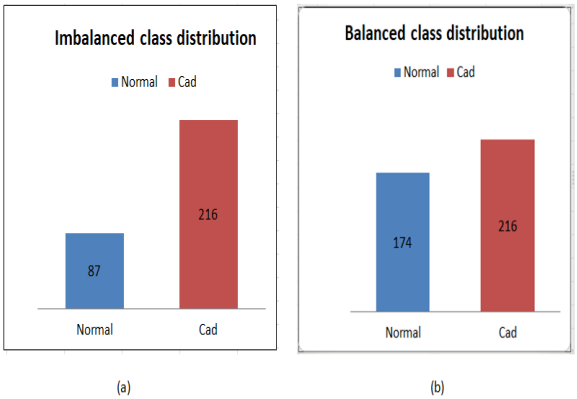



Fig. 2. Class distribution of Normal and Cad

TABLE II. FEATURES OF Z-ALIZADEH SANI DATASET

| Feature type | Feature name | Range |
|---|---|---|
| Demographic | Age | 30-86 |
| | Weight | 48-120 |
| | Length | 140-188 |
| | Sex | Male,Female |
| | BMI(Body Mass Index) | 18.1-40.9 |
| | DM (Diabetes Mellitus) | Yes,No |
| | HTN (Hyper Tension) | Yes,No |
| | Current Smoker | Yes,No |
| | Ex-Smoker | Yes,No |
| | FH (Family History) | Yes,No |
| | Obesity | Yes,No |
| | CRF (Chronic Renal Failure) | Yes,No |
| | CVA (Cerebrovascular Accident) | Yes,No |
| | Airway Disease | Yes,No |
| | Thyroid Disease | Yes,No |
| | CHF (Congestive Heart Failure) | Yes,No |
| | DLP (Dyslipidemia) | Yes,No |
| Clinical | BP (Blood Pressure: mmHg) | 90-190 |
| | PR (Pulse Rate) (ppm) | 50-110 |
| | Edema | Yes,No |
| | Weak peripheral pulse | Yes,No |
| | Lung Rales | Yes,No |
| | Systolic murmur | Yes,No |
| | Diastolic murmur | Yes,No |
| | Typical Chest Pain | Yes,No |
| | Dyspnea | Yes,No |
| | Function Class | 1,2,3,4 |
| | Atypical | Yes,No |
| | Nonanginal | Yes,No |
| | Exertional CP (Exertional Chest Pain) | Yes,No |
| | LowTH Ang (low Threshold angina) | Yes,No |
| | Rhythm | Yes,No |

### *B. Data pre-processing*

In order to address the notable uneven distribution of classes within the dataset, we specify that the (Normal) category pertains to patients who do not have any cardiovascular disease, while the (Cad) category refers to those who do, as illustrated in Fig. 2(a). It should be emphasized that the dataset contains about three times more individuals with CVD than those without it. At this phase, the challenge of class imbalance is resolved by employing the synthetic minority oversampling method (SMOTE). It is an oversampling technique that has gained considerable use in the medical domain for handling imbalanced data [30]. By producing minority class random synthetic data from its closest neighbors using Euclidean distance, SMOTE augments the quantity of data instances. New instances begin to resemble the original data since they are formed based on the original data [31]. A fresh training dataset is created in this work utilizing the SMOTE approach. Each class's data sample size was increased by SMOTE from 303 to 390 as shown in Fig.2 (b). Then, 10-fold cross-validations have been performed to split the dataset into test and train sets.

## V. RESULT AND DISCUSSION

In order to evaluate the effectiveness of the five proposed classifiers for diagnosing cardiac illness, different metrics such as specificity, precision, recall, F1-score, Mcc, and accuracy were utilized. Moreover, we compared the model's results before and after attempted to strike a balance in the dataset.

The findings revealed that certain algorithms demonstrated strong accuracy, while others performed poorly prior to balancing the data through the use of over-sampling. Table 3 displays the performance of the utilized classifiers on the raw data (imbalanced data), demonstrating that SVM had the highest accuracy performance at 86.798% and other metrics.

Table 4 presents the performance of the classifiers on balanced data. A noticeable improvement in the performance of all classifiers across all metrics was observed with 10-fold cross-validation. For instance, RF's accuracy improved from 85% to 90%, DT's accuracy improved from 79% to 84%, LR's accuracy increased from 83% to 86%, SVM's accuracy improved from 86.79% to 87.69%, and MLP's accuracy increased from 82% to 85%.

It is noteworthy that improvements in all evaluation metrics, particularly in MCCs performance, were achieved. Table 5 and Fig. 3 provide a comparison of the accuracy of the ML classifiers before and after balancing the dataset.

Finally, to provide a more comprehensive comparison with previous studies, we discuss studies that have used the imbalanced dataset and the same resampling (SMOT) techniques [9, 11] from Table 1, it is evident that our proposed MDSS has a higher accuracy with 90.51% over algorithms of [9] with 87.6% accuracy. Moreover, our approach outweighed the study [11], although they used more than resampling techniques over the dataset and three classifiers.

TABLE III. ML ALGORITHMS PERFORMANCE MEASURES WITH TEST MODEL 10-FOLD CROSS-VALIDATION (IMBALANCED DATA)

| Classifier | Accuracy | Precision | Recall | F-Measure | MCC |
|---|---|---|---|---|---|
| ***SVM*** | 86.798 % | 0.911 | 0.903 | 0.907 | 0.680 |
| ***RF*** | 85.808% | 0.865 | 0.949 | 0.905 | 0.637 |
| ***DT*** | 79.207 % | 0.837 | 0.880 | 0.858 | 0.474 |
| ***LR*** | 83.168 % | 0.884 | 0.880 | 0.882 | 0.590 |
| ***MLP*** | 82.178 % | 0.879 | 0.870 | 0.874 | 0.568 |

TABLE IV. ML ALGORITHMS PERFORMANCE MEASURES WITH TEST MODEL 10-FOLD CROSS-VALIDATION (BALANCED DATA)

| Classifier | Accuracy | Precision | Recall | F-Measure | MCC |
|---|---|---|---|---|---|
| ***SVM*** | 87.692 % | 0.893 | 0.884 | 0.888 | 0.751 |
| ***RF*** | 90.512% | 0.909 | 0.921 | 0.915 | 0.808 |
| ***DT*** | 84.615% | 0.861 | 0.861 | 0.861 | 0.689 |
| ***LR*** | 86.666 % | 0.887 | 0.870 | 0.879 | 0.731 |
| ***MLP*** | 85.641 % | 0.908 | 0.824 | 0.864 | 0.716 |

TABLE V. COMPARISON IN ACCURACY OF CLASSIFIERS WITH IMBALANCED AND BALANCING

| Classifier | Accuracy with 10-fold cross-validation | |
|---|---|---|
| | Imbalanced dataset | Balanced dataset |
| ***SVM*** | 86.798 % | 87.692 % |
| ***RF*** | 85.808% | 90.512% |
| ***DT*** | 79.207 % | 84.615% |
| ***LR*** | 83.168 % | 86.666 % |
| ***MLP*** | 82.178 % | 85.641 % |

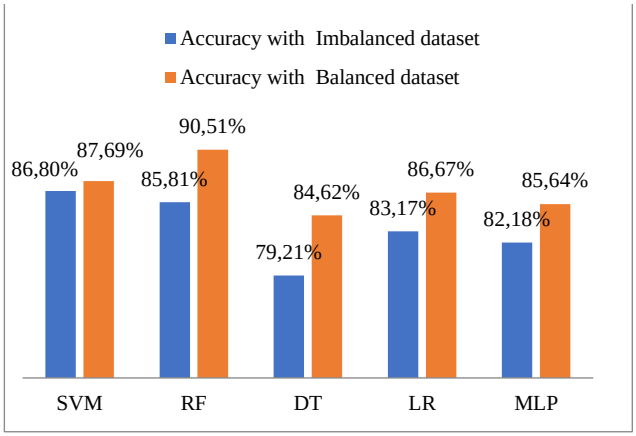



Fig. 3. Accuracy of classifiers with 10-fold cross-validation for imbalanced and balanced dataset

## VI. Conclusion & Future Work

The progress in ML techniques has made it possible to use data mining effectively in healthcare. The study conducted aimed to develop a model for diagnosing heart disease using biochemical values at a lower cost. The results showed that all classifiers had higher accuracy rates when subjected to a 10-fold cross-validation test model after balancing the dataset. The study highlights the importance of addressing the class imbalance in preparing datasets for effective machine learning and presents a methodology for using multiple classifiers to predict CVD. This proposed methodology has the potential to enhance diagnostic accuracy, detect patients at an early stage, decrease mortality rates, and enable further treatment, especially in situations with imbalanced datasets. As technology advances, future studies should aim to expand this approach to larger datasets and leverage deep learning principles. This will allow for even more accurate diagnoses, better patient outcomes, an overall improvement in healthcare services, and an improved quality of life for people worldwide.

## References

[1] Rani, P., Kumar, R., Ahmed, N. M., & Jain, A. (2021). A decision support system for heart disease prediction based upon machine learning. Journal of Reliable Intelligent Environments, 7(3), 263-275.

[2] https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds) ,11 June 2021.

[3] Shah, D., Patel, S., & Bharti, S. K. (2020). Heart disease prediction using machine learning techniques. SN Computer Science, 1(6), 1-6.

[4] Swathy, M., & Saruladha, K. (2021). A comparative study of classification and prediction of Cardio-Vascular Diseases (CVD) using Machine Learning and Deep Learning techniques. ICT Express.

[5] Baghel, N., Dutta, M. K., & Burget, R. (2020). Automatic diagnosis of multiple cardiac diseases from PCG signals using convolutional neural network. Computer Methods and Programs in Biomedicine, 197, 105750.

[6] Nangia, R., Singh, H., & Kaur, K. (2016). Prevalence of cardiovascular disease (CVD) risk factors. medical journal armed forces india, 72(4), 315-319.

[7] Ali, M. M., Paul, B. K., Ahmed, K., Bui, F. M., Quinn, J. M., & Moni, M. A. (2021). Heart disease prediction using supervised machine learning algorithms: performance analysis and comparison. Computers in Biology and Medicine, 136, 104672.

[8] Pavithra, V., & Jayalakshmi, V. (2021). Hybrid feature selection technique for prediction of cardiovascular diseases. Materials Today: Proceedings.

[9] Kolukisa, B., & Bakir-Gungor, B. (2023). Ensemble feature selection and classification methods for machine learning-based coronary artery disease diagnosis. Computer Standards & Interfaces, 84, 103706.

[10] Saboor, A., Usman, M., Ali, S., Samad, A., Abrar, M. F., & Ullah, N. (2022). A Method for Improving Prediction of Human Heart Disease Using Machine Learning Algorithms. Mobile Information Systems, 2022.

[11] Türkmenoğlu, B. K., & Yildiz, O. (2021, June). Predicting the survival of heart failure patients in unbalanced data sets. In 2021 29th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.

[12] Patro, S. P., Nayak, G. S., & Padhy, N. (2021). Heart disease prediction by using novel optimization algorithm: A supervised learning prospective. Informatics in Medicine Unlocked, 26, 100696.

[13] Sudha, V. K., & Kumar, D. (2023). Hybrid CNN and LSTM Network For Heart Disease Prediction. SN Computer Science, 4(2), 172.

[14] Shah, W., Aleem, M., Iqbal, M. A., Islam, M. A., Ahmed, U., Srivastava, G., & Lin, J. C. W. (2021). A Machine-Learning-Based System for Prediction of Cardiovascular and Chronic Respiratory Diseases. Journal of Healthcare Engineering, 2021.

[15] Perumal, R., & Kaladevi, A. C. (2020). Early prediction of coronary heart disease from cleveland dataset using machine learning techniques. Int. J. Adv. Sci. Technol, 29, 4225-4234.

[16] Rajendran, R., & Karthi, A. (2022). Heart disease prediction using entropy based feature engineering and ensembling of machine learning classifiers. Expert Systems with Applications, 207, 117882.

[17] Pan, C., Poddar, A., Mukherjee, R., & Ray, A. K. (2022). Impact of categorical and numerical features in ensemble machine learning frameworks for heart disease prediction. Biomedical Signal Processing and Control, 76, 103666.

[18] Jackins, V., Vimal, S., Kaliappan, M., & Lee, M. Y. (2021). AI-based smart prediction of clinical disease using random forest classifier and Naive Bayes. The Journal of Supercomputing, 77(5), 5198-5219.

[19] Chinnasamy, P., Kumar, S. A., Navya, V., Priya, K. L., & Boddu, S. S. (2022). Machine learning based cardiovascular disease prediction. Materials Today: Proceedings.

[20] Gao, C., & Elzarka, H. (2021). The use of decision tree based predictive models for improving the culvert inspection process. Advanced Engineering Informatics, 47, 101203.

[21] Schober, P., & Vetter, T. R. (2021). Logistic regression in medical research. Anesthesia and analgesia, 132(2), 365.

[22] Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing, 408, 189-215.

[23] Sheykhmousa, M., Mahdianpari, M., Ghanbari, H., Mohammadimanesh, F., Ghamisi, P., & Homayouni, S. (2020). Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13, 6308-6325.

[24] Valupadasu, R., & Chunduri, B. R. R. (2019, May). Automatic classification of cardiac disorders using MLP algorithm. In 2019 Prognostics and System Health Management Conference (PHM-Paris) (pp. 253-257). IEEE.

[25] Juba, B., & Le, H. S. (2019, July). Precision-recall versus accuracy and the role of large data sets. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 4039-4048).

[26] Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061.

[27] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC genomics, 21(1), 1-13.

[28] Chicco, D., Tötsch, N., & Jurman, G. (2021). The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. BioData mining, 14(1), 1-22.

[29] https://archive.ics.uci.edu/ml/datasets/Z-Alizadeh+Sani.

[30] Blagus, R., & Lusa, L. (2015). Joint use of over-and under-sampling techniques and cross-validation for the development and assessment of prediction models. BMC bioinformatics, 16(1), 1-10.

[31] Kovács, G. (2019). An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets. Applied Soft Computing, 83, 105662

# A Short Review: Cyber Attacks And Detection Methods Based On Machine Learning And Deep Learning Approaches In Smart Grid

*Received: 1 January 2023; Accepted: 6 March 2023* *Review Article*

Mehmet KARAYEL
Computer Engineering
Kocaeli University
Kocaeli, Türkiye
226112004@kocaeli.edu.tr

Nevcihan DURU
Faculty of Engineering and Natural Sciences
Kocaeli Health and Technology University
Kocaeli, Türkiye
nevcihan.duru@kocaelisaglik.edu.tr

Mehmet KARA
Faculty of Engineering and Natural Sciences
Kocaeli Health and Technology University
Kocaeli, Türkiye
0000-0001-7312-0503

***Abstract*—Power systems and smart grids constitute critical instruments of national security and the economy. In case of the power system malfunctioning, millions of people are affected. Furthermore, there are extreme financial losses, irreversible data casualties and service outages. Recently, the use of commercial smart measuring and control devices in the field of electricity and power systems has become widespread due to the development of applicable technologies and the reduction of the costs of devices. Although this situation has increased traceability and manageability, it also made smart grids more vulnerable to cyber threats compared to the traditional power systems used before. Cyber threats in smart grids are generally categorized as eavesdropping the data to possess detailed information about the system, tampering with data to disturb the system's stability, denial of services to block accessibility and injecting malicious software that can cause damage to the system. FDI attack is considered one of the most severe cyber-attack types due to its stealthy. FDI attacks disrupt the entire stabilization of the smart grid gradually. Machine learning and deep learning methods in supervised, semi-supervised and unsupervised domains have been widely used to protect smart grids against cyber threats by assisting conventional bad data detection mechanisms. Successful results have mainly been obtained by deep learning algorithms such as CNN and RNN. These algorithms have been supported with improved feature selection techniques to increase the accuracy of the detection and decrease the computational burden of the models. The purpose of the paper is to briefly summarize and combine the significance of smart grids, vulnerabilities of smart grids, cyber threats to smart grids, deep learning and machine learning methods applied against cyber-attacks, especially FDI attacks considered to be the most dangerous attack type and potential future research areas.**

***Keywords*—Power Systems, Smart Grids, Cyber Attack, False Data Injection Attack, FDIA, Machine Learning, Deep Learning, CNN, RNN, LSTM.**

## I. INTRODUCTION

As a result of the introduction of Industry 4.0 [1] and Industry 5.0 [2] with the development of technology, human-machine interactions have started to appear more in every field than ever before since 2010. In addition to the favorable benefits brought by these technologies, the energy demand has increased dramatically. At the same time, energy continuity and supply-demand balance are critical parameters that all countries and companies must monitor because a failure or any problem in these systems affects all infrastructure. Therefore, energy generation and distribution systems are at the forefront of critical infrastructures.

Considering that the resources are not infinite, energy demand triggers the finding of solutions for using the energy more efficiently in a controlled environment at an optimum level. At this point, smart grids that meet energy demands intelligently come into play with new features and capabilities compared to traditional electricity grids in terms of traceability and manageability.

Power systems and smart grids are becoming critical infrastructures in recent years. The dependency on the power system and smart grid is increasing rapidly. Many people are affected by any problem in the power system or smart grid, and the losses are very high in the cost and information domain. It is considered that smart grids are still vulnerable to cyber-attacks since they are an extension of legacy systems, supported by commercial devices with no advanced security infrastructure and a lack of security protocols already existing in other networks like the internet.

The rest of the paper is organized as follows: Section 2 presents the grid conceptual model and architecture including vulnerabilities and cyber-attack types to smart grids. Detection Methods of FDI attacks are detailed based on machine learning and deep learning algorithms in section 3. Finally, conclusions and potential research areas are supported in section 4.

## II. SMART GRIDS AND CYBER ATTACKS

Smart grids are sophisticated systems of legacy grids with improved properties. They combine different types of electricity production (Distributed Energy Resources-DER), like solar power, wind power, hydroelectric, etc., in one framework. Smart grids comprise all processes from production to consumption of electricity. Smart grids have been made intelligent by Information and Communication Technologies (ICT) such as control panels, sensors, actuators, measuring devices and smart meters. The most important feature that distinguishes smart grids from traditional systems is that instead of transmitting electricity in one-way, communication and power flows are conducted in two-way.

Smart grids have transformed into highly complex structures due to integrated information and communication systems. To minimize the complexity and clarify standards, a conceptual model was initially proposed in 2010 [3]. The

conceptual model has been revised periodically based on recent developments. The up-to-date conceptual model [4] is depicted in Figure 1. In general, a smart grid consists of 7 main domains. Data and power flow are carried out between the Generation, Transmission, Distribution and Consumption domains. On the other hand, data transmission is conducted between Operation, Service Provision and Market domains.

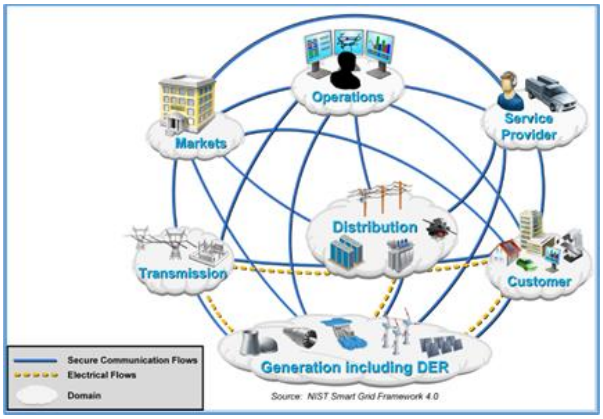



Fig. 1. The up-to-date NIST Smart Grid Conceptual Model.

### A. *Security Vulnerabilities of Smart Grids*

Smart Grids uses Information and Communication Technologies (ICT) infrastructure to manage and monitor the system. ICT systems are vulnerable to cyber-attacks such as False Data Injections (FDI), Denial of Service (DoS), data sniffing, unauthorized access and password cracking. Smart grids are the target of ICS attacks as well as attacks against information systems. Therefore, the attack space is wider than the applied only in IT systems.

ICS systems are very different from Information systems in terms of performance, availability (reliability), risk management, system operation, resource constraints, communications, change management, support and component locations. As it can be understood from this structure, it is more challenging to control ICS system vulnerabilities compared to IT Systems only [5]. Vulnerabilities of smart grids can be grouped under the following main headings.

*1) Physical Components Vulnerabilities*: Hardware, software, and management systems make up a smart grid. However, they each have some vulnerabilities, such as insufficient physical access control, redundancy, component maintenance, and HAVC (Heating, Ventilation, and Air Conditioning) systems [6].

*2) IT Vulnerabilities*: One of the main functions of IT systems is automating business functions like billing, customer service, and accounting. Since the commence of the internet, IT systems have been employed, and a wealth of knowledge has been gained about them. There are many vulnerabilities, such as insecure software and hardware, confidentiality issues, integrity troubles and authentication and authorization problems. On the other hand, many critical areas, such as e-government, e-banking and e-commerce systems, are operated securely [7].

*3) OT Vulnerabilities*: A major focus of OT has been the management of power system operations, such as distribution of power and critical energy infrastructure management. OT advancements have led to more automated substations that can operate without human interaction. Software vulnerabilities in measurement devices, such as HMI (Human Machine Interface), RTU (Remote Terminal Unit), sensors and actuators, should be considered significantly critical matters which can lead to destructive cyber-attacks [8,9].

*4) Data Processing and Management Vulnerabilities*: Current smart grid data management faces the problem of data aggregation quality, security, compliance control, typical scope, and efficiency of the management mechanism. Many data are generated, processed, stored and transferred between different entities. The Confidentiality, Integrity and Availability of this data must be the focal points and be protected strictly. Data security and privacy should always be the priority in the design of smart grids [10].

*5) Service and Application Vulnerabilities*: The instant conversion of physical data into useful information is made possible by access to OT and IT data, enabling enhanced asset management platforms, distributed energy management systems, and distribution grid applications. Electricity trading, electricity services, electricity convergence, and a variety of client services are just a few of the applications and services that smart grids can offer. These services may include patching, policy, asset management, configuration, authentication, authorization, accounting and malware vulnerabilities.

*6) Operational Environment Vulnerabilities*: Unknowing employees, poor outsourcing, insecure configuration, and issues with the natural environment are some of the common risks for the operating environment of the grid.

### B. *Cyber Attacks to Smart Grid*

Since 2010, there have been many examples that ended with financial losses and physical damage around the world. The most effective of all, attacks targeting Iran's nuclear facilities and Ukrainian power systems come to the fore. STUXNET targeted SCADA (Supervisory Control and Data Acquisition) systems and caused substantial damage to Iran's nuclear program, including the nuclear centrifuge, computers and ICS devices [11]. In 2016, a power system outage in Ukraine affected many part of the country and many customers [12].

Smart grid attacks are seen in a wide range, such as FDI attacks, denial of service (DoS) attacks, data framing attacks, man-in-the-middle attack, load altering attacks, false command injection attacks, load redistribution attacks, coordinated cyber–physical topology (CCPT) attacks and replay attack. These attacks can be grouped as IT-based, ICS-based and grid data based attacks. There are deep knowledge and preventive tools for IT-based attacks. Relatively less knowledge and preventive tools on ICS-based attacks and smart grid data based attacks.

As stated in the previous subsection, many critical areas, such as e-government, e-banking and e-commerce systems, can be operated safely in today's conditions where cyber-attacks are assumed and accepted. The infrastructure can be easily changed and adapted to new situations using simple costs in the mentioned areas. However, the situation is different in power systems and smart grids. As mentioned, power systems are being made smart by integrating commercial sensors and measurement systems of existing legacy systems. Therefore, advanced secure communication protocols and defense systems used on the internet are not used in power systems and smart grids. Artificial intelligence

methods are included in the game precisely at this point. Artificial intelligence fills the gap in the need for advanced security systems in smart grids.

FDI attack is considered one of the most severe cyber-attack types among the cyber-attacks mentioned above due to its stealthy. FDI attacks in a smart grid first appeared in [13]. They disrupt the entire stabilization of the smart grid gradually. Recently, FDI attacks received noticeable attention due to their impact. State estimation plays a critical role in the stable operation of smart grids at an optimum level. FDI attacks directly target state estimation. Failure to perform the state estimation process properly causes enormous damage and power outages. Because FDI attacks can be made relatively easily, but their damage to smart grids can be comprehensive, the methods applied against cyber-attacks are explained based on FDI attacks in the following sections.

## III. Detection Of Cyber Attacks Using Machine Learning And Deep Learning Approaches

In the literature, studies were firstly based on simple and effective machine learning methods like Decision trees, Random Forests, etc., due to their simplicity and computational efficiency. After the deep learning methods became popular in other domains like image recognition and classification, deep learning practices such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and LSTM (Long-Short Term Memory) techniques were also widely applied for the detection of cyber-attacks in smart grids. Semi-supervised and unsupervised learning approaches are popularly used in this field, as accurate data on cyber-attacks in smart grids are rare and labelling datasets is highly time-consuming. Generally, in semi-supervised learning, unlabeled data is assigned to the nearest neighbor data set based on the labeled classes by kernel methods such as RBF (Radial Basis Function) or KNN (K Nearest Neighbor) [14]. In addition, to obtain more realistic training examples, a more advanced and complex model named Generative Adversarial Neural Network (GAN) is started to be used recently [15]. In the GAN model, labeled data are produced by two separate but linked Neural Networks, a generator and a discriminator, with feedback in an iterative min-max game.

Unlike supervised and unsupervised learning methods, fully unlabeled data is used to train the model in unsupervised approaches. Unsupervised methods such as Principal Component Analysis (PCA) and KNN-based methods are primarily and widely used in the literature [16].

Furthermore, the approaches to detecting cyber-attacks are categorized according to whether they depend on a model [17]. State estimation techniques in model-dependent and comparison of sequential temporal data in model-independent approaches are used based on the data collected during the system's regular operation. FDI attacks are targeted at the measurement data via tampering with the measurements to deceive the system. Traditional Bad Data Detectors can only be adapted for outlier information like false reading and cannot perceive hidden interventions like FDI attacks.

The most commonly applied Machine Learning and Deep Learning algorithms used to detect cyber-attacks in smart grids are summarized in Table I. Furthermore, the main strengths and weaknesses of the algorithms are indicated in terms of reinforcing knowledge about algorithms in Table II.

The strengths and weaknesses of the algorithms are assessed within the framework of general performance criteria. The assessments are made in terms of the internal structure of data, the data preprocessing phase, applying methods of algorithms, interior design and objectives of the algorithms, the algorithms' parameters, the algorithms' performance, the training phase of the algorithm, etc. In addition, only the most prominent features are highlighted in Table II.

TABLE I. Most commonly applied Machine Learning and Deep Learning algorithms

| Learning Types | List of Algorithms |
|---|---|
| Machine Learning Algorithms | SVM (Support Vector Machine), KNN (K-Nearest Neighbor), ENN (Extended Nearest Neighbor), PCA (Principal Component Analysis), AdaBoost, Decision Tree, Random Forest, LGBM (Light Gradient Boosting Machine) and Logistic Regression. |
| Deep Learning Algorithms | ANN (Artificial Neural Networks), CNN (Convolutional Neural Networks), RNN (Recurrent Neural Networks), LSTM (Long-Short Term Memory) and GAN (Generative Adversarial Neural Networks). |

TABLE II. Main strengths and weaknesses of the algorithms

| Algorithm | Strengths | Weaknesses |
|---|---|---|
| SVM | 1. Satisfactory performance in high dimensional space.<br>2. Outliers have a minor impact. | Slow training process for large datasets. |
| KNN | 1. Simple to implement.<br>2. No presumptions about data. | Sensitive to outliers. |
| ENN | It can learn from the global distribution in addition to local one used in KNN. | Choosing of parameter “K” like KNN. |
| PCA | It reduces overfitting. | There is a possibility of information loss. |
| AdaBoost | It can be slightly less sensitive to overfitting. | It is susceptible to noisy data and outliers. |
| Decision Tree | Normalization or scaling of data not needed. | Prone to overfitting. |
| Random Forest | Promising performance on unbalanced and missing data. | It requires much computational power and time. |
| LGBM | Reduced training time and low memory usage. | It needs much complex trees. |
| Logistic Regression | Tuning of hyperparameters is not needed. | Inadequate performance on nonlinear data. |
| ANN | Suitable for modelling nonlinear data with a higher dimension. | There is no exact rule for defining the structure of the network. |
| CNN | It detects critical features in an unsupervised manner. | Training process may take considerable time depending on the number of network layers. |
| RNN | It is the first neural network able to analyse and learn sequences of data (series) of its kind. | Vanishing gradients. |
| LSTM | LSTMs are forceful RNNs designed to work with vanished gradients. | The training data required by LSTMs is much greater than that needed by CNNs and RNNs to achieve the same level of accuracy. |
| GAN | It can generate data similar to the original in an unsupervised manner. | Having two separate networks, a generator and a discriminator, makes training phase difficult. |

It is evaluated that it would be helpful to explain the concept of "State Estimation" and "Bad Data Detector" approaches, which constitute the main pillars of controlling and protecting the smart grids, are in the stage before the detection algorithms are applied. It also explains how FDI attacks are theoretically produced and why the BDD cannot recognize them.

### A. *State Estimation, Bad Data Detection (BDD) and False Data Injection Attacks*

The main goal of state estimation is to predict a smart grid's current state using sensors' data. The measurements usually consist of real and reactive power injections of transmission lines and buses and state variables like all buses' voltage magnitudes and phase angles. In the state estimation, the relationship is conducted based on (1) between the state vector $x \in \mathbb{R}^D$ and measurements $z \in \mathbb{R}^N$. In addition, $H \in \mathbb{R}^{NxD}$ is Jacobian topological matrix and **e** is the error.

$$z = Hx + \mathrm{e} \tag{1}$$

The state of the smart grid can be predicted by Weighted Least Square (WLS) where $W$ is a diagonal matrix with elements proportional to the variance of each measurement noise, defined as follows :

$$\hat{x} = arg_x\, min(z - Hx)^T W (z - Hx) \tag{2}$$

To eliminate the measurement error and sensor faults, BDD is applied as a first defensive mechanism to protect the state estimation. In the traditional BDD, the L2-norm of measurement residual is compared to the threshold $\tau$ and measurement data is not accepted when $\|z - \mathrm{H}\hat{x}\| > \tau$ .

FDI attacks are created by adding an attack vector $a$ to the measurement vector $z$. So state variable vector is transformed into (4) where $c \in \mathbb{R}^N$ is the difference in the state variable estimates. When the attack vector content with (5), then the L2 norm of attacked measurement is defined in (6). Equation 6 shows that L2 norm of attacked measurement residual does not change. It means that attacked measurements can bypass the BDD.

$$z_a = z + \mathrm{a} \tag{3}$$

$$\hat{x}_a = \hat{x} + \mathrm{c} \tag{4}$$

$$a = Hc \tag{5}$$

$$\|z_a - H\hat{x}_a\| = \|z + \mathrm{a} - \mathrm{H}(\hat{x} + \mathrm{c})\| \tag{6}$$

$$= \|z - \mathrm{H}\hat{x} + (\mathrm{a} - \mathrm{Hc})\|$$

$$= \|z - \mathrm{H}\hat{x}\|$$

### B. *Efficient Legacy Machine Learning Algorithms*

The previous subsection explains the concepts of state estimation and BDD techniques, which indicate the current status of smart grids and lay the foundation for detection algorithms to be applied to detect cyber-attacks. This subsection details the recently applied machine learning methods for detecting cyber-attacks.

KNN algorithm, suitable for classification and regression purposes, is used as a main algorithm in [18, 19]. A Robust KNN regression approach is proposed in [18] to eliminate uncertainties and conduct more accurate state estimation in power systems. Furthermore, combining the KNN algorithm with PCA (Principal Component Analysis) feature selection based on the critical concept feature set is implemented in [19]. In addition to the studies in which KNN is the primary classifier, it is noteworthy that in other studies, it is usually in the domain of the compared algorithms. In [20], three common classifiers like SVM (Support Vector Machine), KNN and ENN (Extended Nearest Neighbor) algorithms were used for FDIA detection and SVM performed superior overall compared to KNN and ENN classifiers.

SVM, which avoids the difficulties of using linear functions in the high-dimensional feature space, is applied to detect cyber-attacks in smart grid that has a non-linear component in nature. In [21], an updated SVM-based method is proposed to detect the FDIA, bringing the vulnerabilities of existing SVM-based FDI attack detectors forward. Furthermore, a detection framework with an SVM classifier at its core with edge data aggregators to detect FDI attacks on transmission lines in [22].

Statistical models that prioritize catching the uncertainty in the models are used to detect the anomalies caused by FDI (False Data Injection) attacks. The multivariate Gaussian model improved with the k-means clustering method for detecting transient and steady cyber-attacks implemented in [23]. It is assumed that the multivariate Gaussian model captures the correlations between variables from different dimensions. In [24], a strategy based on statistical features is put forth to locate supervised FDI attacks in power grids. This method includes quantification of the distribution of the measurements and the tree boosting technique.

Random Forest algorithm, which can deal with both categorical and continuous variables efficiently, is also applied in this field. In [25], various classification methods, like Naive Bayes, Random Forest, etc., are used for detecting power system anomalies. The Random Forest algorithm gets the highest score among other methods. Furthermore, Isolation Forest Algorithm proposed with some acceptance criteria based on the Random Forest method in an unsupervised way in [26].

PCA technique which removes correlated features and reduces overfitting is commonly applied to improve the performance of models. In [27], high-dimensional space is reduced by KPCA (Kernel PCA) and the Extra-Trees algorithm is used to classify stealthy cyber-attacks. KPCA-supported Extra-Trees based detection approach outperforms the state-of-art machine learning-based schemes. Furthermore, [28] proposes a method for FDI attack detection based on PCA and subspace analysis utilizing consequential grid states.

Feature engineering aims to prepare an input dataset that best fits the machine learning algorithm and enhances the models' performance. Optimization algorithms and heuristic algorithms are commonly used. SVM is proposed as a primary classifier and compared to the other machine learning methods like AdaBoost and KNN to detect covert cyber deception assault attacks. Various feature selection methods including GA (Genetic Algorithm) are implemented. As a result, SVM has more successful results than other implemented algorithms [29]. In [30], SVM and KNN algorithms with three different feature selection methods, such as BCS (Binary Cuckoo Search), BPSO (Binary Particle Swarm Optimization) and GA

(Genetic algorithm), are studied to detect the FDI attacks. SVM and KNN algorithms performed more accurately compared to others. Furthermore, the AdaBoost classifier supported with Random Forest is used as the main of the proposed model with the enhanced feature construction engineering techniques in [31].

In [32], a framework combination of a square-root unscented Kalman filter (SR-UKF) based forecasting-aided State Estimation and a GLRT (Generalized Likelihood Ratio Test) is designed to detect FDI attacks in unbalanced distribution networks.

It is determined that SVM and KNN are applied much more than other machine learning algorithms to detect FDI attacks. Considering the complexity and non-linearity structure of the data obtained in smart grids, it has been observed that the SVM algorithm, built on the theory of creating a hyperplane, exhibits very reliable performances. Furthermore, the KNN algorithm is significantly utilized because it has a non-parametric and straightforward structure. In addition, KNN is ideal for non-linear data since there is no assumption about underlying data.

### *C. Deep Learning Algorithms*

Convolutional Neural Network (CNN) is an artificial neural network architecture for deep learning with fully connected input, convolution, pooling and output layers that learn directly from data. Since the algorithm is based on learning directly from the data, invisible patterns can be revealed. Besides, RNN is a particular variant of ANN (Artificial Neural Network) for analyzing sequential data. Furthermore, the Long Short-Term Memory (LSTM) algorithm is an extension of RNN that extend the memory to eliminate the short-term memory. LSTM models can retain past information even longer compared to RNN algorithms.

Deep learning algorithms come to the fore in processing big data produced within the scope of monitoring and controlling smart grids to detect cyber-attacks. To facilitate big data processing, autoencoders within deep learning algorithms reduce the dimensionality of data and the computational complexity [33-35]. Furthermore, by using traditional machine learning algorithms and deep learning methods together, nonlinear data have been transformed into linear space and the detection accuracy of cyber-attacks has been increased [36].

Recently, deep learning algorithms combined with traditional machine learning algorithms such as SVM and KNN have been commonly used to detect cyber-attacks [37]. Furthermore, among the deep learning methods, LSTM, RNN and CNN are considered the most used and successful algorithms [38].

In [39], some ML-based models, such as SVM and Light Gradient Boosting Machine (LGBM) are compared with Deep Learning (DL) based models like CNN and ANN. As a result of the experiments, it has been determined that CNN models give better results. In [40], Continuous wavelet transform (CWT) is used to transform one-dimensional traffic data into a two-dimensional time-frequency domain as input to a wavelet CNN (WavCovNet) to distinguish the cyber-attack and detect abnormal behavior in the data.

Considering that smart grid data naturally contains linear and nonlinear components, an effective two-level FDIA detection is performed using the Kalman filter and RNN (KFRNN) [41]. In the first stage, the Kalman filter is used for state estimation from linear data and RNN is used to capture nonlinear data features. At the second level, the results obtained from the processes of linear and nonlinear are fed into a fully connected neural network with backpropagation. Furthermore, taking into account the same assumption accepted in [41] regarding linear and nonlinear components, RNN has also been proposed to detect the FDI attacks in [42].

Furthermore, [43] and [44] provide examples of how time series measurement values are effectively handled with RNN-based models. In [45], time domain data were processed with the LSTM autoencoder and anomalies are detected using the Logistic Regression. In [46], the state estimation to detect cyber-attacks is conducted through a model obtained by combining multiple LSTMs. In [47], the proposed framework is based on consolidating the Wasserstein Generative Adversarial Network and autoencoder to learn the smart grid measurement distribution and state estimator model.

As can be understood from the studies described above, deep learning algorithms such as CNN, RNN and LSTM and structures that combine these algorithms with traditional machine learning methods are widely and effectively used in detecting FDI attacks based on state estimation or time series. Furthermore, promising results have been obtained by using deep learning and machine learning algorithms together.

### *D. Simulations and Datasets*

It would be helpful to specify the issue of obtaining the data. Since datasets regarding actual cyber-attacks are confidential and almost inaccessible, simulation methods are widely used to generate training and test data. The regular operation of the smart grid is simulated like a real-time environment to obtain the data. Furthermore, FDI attacks are implemented via tampering with the generated data.

In most of the studies, simulations are commonly conducted using the MATPOWER simulation package [48]. In addition, most of the simulations are performed based on IEEE Bus Systems Data. These systems consist of loads, capacitor banks, transmission lines and generators and their reference values.

### *E. Main Issues and Future Directions*

Cyber-attacks against smart grids can be roughly grouped as obtaining individual or system data illegally, creating large-scale denial of services, and sabotaging the system using false data. Considering the characteristics of cyber-attacks and instances encountered in real life over the last 20 years, cyber-attacks can be exploited as destructive weapons.

Furthermore, cyber-attacks are highly concentrated in critical infrastructure areas such as energy generation, especially nuclear infrastructure, the nation's military and civilian defense systems, bank and finance systems, communication systems, logistic and critical commercial port systems and services.

In the last 20 years, the main issues that drive using of artificial intelligence solutions to defend smart grids are as follows:

- Power systems and intelligent grids become the most valuable resources of a nation,

- Confidentiality, integrity and availability of the information produced in power systems and smart grids is vital,
- The need for controlling and monitoring activities in power systems and smart grids,
- Easy access to the commercial measuring systems and control equipment and vulnerabilities of commercial devices,
- Limited resources and the need to use energy efficiently at the maximum level,
- Increasing self-operating activities in every field and the inevitability of automation in the Industrial 5.0 era,
- Most of the ICTs used in the energy generation area are dependent on obsolete technology, so unlike internet infrastructure,

The following items can serve as the basis for future studies and research to ensure power systems and smart grids have a higher level of security against various cyber-attacks and provide uninterrupted operation.

- Developing new frameworks for the detection of multiple cyber-attacks at the same time,
- A deeper understanding of determining fingerprints and features of cyber-attacks,
- Building up new cascaded frameworks applying more distributed controlling and detecting manners over the whole system,
- Creation of cyber-attacks more realistically with various deep learning algorithms in addition to the existing GAN algorithm to develop sound defense systems due to the rarity of real datasets related to the cyber-attacks.

## IV. CONCLUSION

In this study, the structure and working principles of smart grids, the vulnerabilities of smart grids, the types of cyber-attacks against smart grids, the most harmful FDI attacks, the reasons behind why machine and deep learning algorithms are needed in smart grids, the idea of how detection algorithms are used to detect cyber-attacks, the simulation and datasets used in studies in this field are shortly reviewed.

It is observed that traditional machine learning and deep learning algorithms have been successfully applied alone in detecting cyber-attacks, and promising results have been obtained. In addition, it should be mentioned that machine learning and deep learning methods are used together, and satisfactory results are obtained. Furthermore, it has been determined that SVM and KNN out of machine learning algorithms and CNN, RNN and LSTM from deep learning are the most used methods in detecting cyber-attacks in the literature.

During the literature review, it was observed that detection models were generally developed against a single attack type. The development of models for detecting two different types of attacks by obtaining the similar feature set between attack types and conducting detections based on using this similarity can shed light on potential future studies.

## REFERENCES


[1] S. Vaidya, P. Ambad, C. O'Fallon, S. Bhosle, "Industry 4.0 – A Glimpse," 2 nd International Conference on Materials Manufacturing and Design Engineering. 11-12 December 2017, Procedia Manufacturing 20 (2018) 1233-238, 2018.

[2] A.S. George and A.S.H. George, "Industrial Revolution 5.0: The Transformation of The Modern Manufacturing Process To Enable Man And Machine To Work Hand In Hand," Seybold Report. ISSN NO: 1533-9211. September 2020,

[3] GW. Arnold, DA. Wollman, GJ. FitzPatrick, D. Prochaska, DG. Holmberg, DH. Su, AR. Hefner, NT. Golmie, TL. Brewer, and M. Bello "2010 NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0," National Institute of Standards and Technology, Gaithersburg, MD, NIST SP, 1108, 2010.

[4] A. Gopstein, C. Nguyen, C. O'Fallon, N. Hastings, and D. Wollman, "NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 4.0," NIST Special Publication 1108r4, 44(16), 3111-3123, 2021.

[5] K. Keith Stouffer and M. Pease, Guide to Operational Technology (OT) Security, NIST Publication, 2022.

[6] J. Xie, A. Stefanov, and C.C. Liu, Physical and Cybersecurity in a Smart Grid Environment. In Advances in Energy Systems: The Large-Scale Renewable Energy Integration Challenge, Wiley: Hoboken, NJ, USA, 2019, pp. 85–109.

[7] C.M. Mathas, C. Vassilakis, N. Kolokotronis, C.C. Zarakovitis, and M.A. Kourtis, On the Design of IoT Security: Analysis of Software Vulnerabilities for Smart Grids. Energies 2021, 14, 2818.

[8] Y. Xu, Y. Yang, T. Li, J. Ju, and Q. Wang, "Review on cyber vulnerabilities of communication protocols in industrial control system," in Proceedings of the 2017 IEEE Conference on Energy Internet and Energy System Integration (EI2), Beijing, China, 26–28 November 2017.

[9] J. Lazaro, A. Astarloa, M. Rodríguez, U. Bidarte and J. Jimenez, A Survey on Vulnerabilities and Countermeasures in the Communications of the Smart Grid. Electronics, 10, 1881, 2021.

[10] M.Z. Gunduz and R. Das, "Cyber-security on smart grid: Threats and potential solutions," Comput. Network, vol. 169, 107094, 2020.

[11] S. Kriaa, M. Bouissou, and L. Pi`etre-Cambac´ed`es, "Modeling the stuxnet attack with bdmp: Towards more formal risk assessments," in 2012 7th International Conference on Risks and Security of Internet and Systems (CRiSIS), pages 1–8, 2012.

[12] R. Khan, P. Maynard, K. McLaughlin, D. Laverty, and S. Sezer. "Threat analysis of blackenergy malware for synchrophasor based real-time control and monitoring in smart grid," 4th International Symposium for ICS & SCADA Cyber Security Research 2016 4, pages 53–63, 2016.

[13] Y. Liu, P. Ning, and M.K. Reiter, "False data injection attacks against state estimation in electric power grids," Proceedings of the 16th ACM Conference on Computer and Communications Security, CCS 2009, pp. 21–32. ACM, New York, NY, USA, 2009.

[14] S. Sharma, K. R. Niazi, K. Verma, and T. Rawat, "An efficient optimization approach for coordination of network reconfiguration and pv generation on performance improvement of distribution system," in Control Applications in Modern Power System. Springer, pp.269-278, 2021.

[15] Y. Zhang, J. Wang, and B. Chen. "Detecting false data injection attacks in smart grids: A semi-supervised deep learning approach," IEEE Transactions on Smart Grid, 12(1):623–634, 2021.

[16] A. S. Musleh, M. Debouza, H. M. Khalid, and A. Al-Durra, "Detection of False Data Injection Attacks in Smart Grids: A Real-Time Principle Component Analysis," 45th *Annual Conference of the IEEE Industrial Electronics Society*, October 2019.

[17] F. Mohammadi, M. Saif, M. Ahmadi, and B. Shafai. "A Review of Cyber Resilient Smart Grid," 2022 World Automation Congress (WAC), Hybrid, San Antonio, TX, USA, October 11-15 2022.

[18] Yang Weng, Rohit Negi, Christos Faloutsos, and Marija D. Ili´c. Robust data-driven state estimation for smart grid. IEEE Transactions on Smart Grid, 8(4):1956–1967, 2017.

[19] M. Mohammadpourfard, Y. Weng, M. Pechenizkiy, M. Tajdinian, and B. Mohammadi-Ivatloo, "Ensuring cybersecurity of smart grid against data integrity attacks under concept drift", Int. J. Electr. Power Energy Syst., vol. 119, p. 105947, Jul. 2020.

[20] J. Yan, B. Tang, and H. He. "Detection of false data attacks in smart grid with supervised learning," 2016 International Joint Conference on Neural Networks (IJCNN), pages 1395–1402, 2016.

[21] B. Wang, P. Zhu, Y. Chen, P. Xun, and Z. Zhang, "False Data Injection Attack Based on Hyperplane Migration of Support Vector Machine in Transmission Network of the Smart Grid," Symmetry, Vol. 10(5), May 2018.

[22] P. Xun, P. Zhu, Z. Zhang, P. Cui, and Y. Xiong, "Detectors on Edge Nodes against False Data Injection on Transmission Lines of Smart Grid," *Electronics*, Vol. 7(6), Jun. 2018.

[23] Y. An and D. Liu, "Multivariate Gaussian-Based False Data Detection against Cyber-Attacks," IEEE Access, vol. 7, pp. 119804–119812, 2019.

[24] J. Jiang, J. Wu, C. Long, and S. Li, "Location of False Data Injection Attacks in Power System," *2019 Chinese Control Conference*, Jul. 2019.

[25] M. Panthi, "Anomaly detection in smart grids using machine learning techniques," 2020 First International Conference on Power, Control and Computing Technologies (ICPC2T), pages 220–222, 2020.

[26] S. Ahmed, Y. Lee, S. H. Hyun, and I. Koo, "Unsupervised Machine Learning-Based Detection of Covert Data Integrity Assault in Smart Grid Networks Utilizing Isolation Forest," IEEE Trans. Inf. Forensics Secur., vol. 14, no. 10, pp. 2765–2777, October 2019.

[27] M. R. Camana Acosta, S. Ahmed, C.E. Garcia, and I. Koo, "Extremely randomized trees-based scheme for stealthy cyber-attack detection in smart grid networks," IEEE Access, vol. 8, pp. 19921–19933, 2020.

[28] E. Drayer and T. Routtenberg, "Intrusion Detection in Smart Grid Measurement Infrastructures Based on Principal Component Analysis," *2019 IEEE Milan PowerTech*, Jun. 2019.

[29] A. Saeed, L. Youngdoo, H. Seung-Ho, and I. Koo. "Feature selection–based detection of covert cyber deception assaults in smart grid communications networks using machine learning," IEEE Access, 6:27518–27529, 2018.

[30] J. Sakhnini, H. Karimipour and A. Dehghantanha, "Smart Grid Cyber Attacks Detection Using Supervised Learning and Heuristic Feature Selection," 2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE), Oshawa, ON, Canada, pp. 108-112, 2019.

[31] D. Wang, X. Wang, Y. Zhang, and L. Jin, "Detection of power grid disturbances and cyber-attacks based on machine learning", J. Inf. Secur. Appl., vol. 46, pp. 42–52, Jun. 2019.

[32] S. Wei, J. Xu, Z. Wu, Q. Hu, and X. Yu, "A False Data Injection Attack Detection Strategy for Unbalanced Distribution Networks State Estimation," IEEE Transactions on Smart Grid, in press.

[33] S.H. Majidi, S. Hadayeghparast, and H. Karimipour, H. "FDI attack detection using extra trees algorithm and deep learning algorithm-autoencoder in smart grid," International Journal of Critical Infrastructure Protection, 2022.

[34] J. Ding, A. Qammar, Z. Zhang, and H. Ning, "Cyber Threats to Smart Grids:Review, Taxonomy, Potential Solution and Future Directions," MDPI, 2022.

[35] L. Gotsev, B. Jekov, Y., Parusheva, and E. Kovatcheva, "Cyber Threats on Smart Grid: Concerns, Attacks and Advanced Detection," 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET), 2022.

[36] T. Teng and L. Ma, "Deep learning-based risk management of financial market in smart grid, " Computers and Electrical Engineering, 2022.

[37] D.M., Menon and N.A. Radhika, "Trust-Based Framework and Deep Learning-Based Attack Detection for Smart Grid Home Area Network.", International Journal of Intelligent Engineering and Systems, 2022.

[38] R. Rituraj, "An Investigation into Methods and Applications of Deep Learning in Smart Grid," ICCC 2022, 10th Jubilee International Conference on Computational Cybernetics and Cyber Medical Systems, 2022.

[39] A. Khan, "Detection of False Data Injection Cyber-Attack in Smart Grid by Convolutional Neural Network-Based Deep Learning Technique," Lecture Notes in Electrical Engineering, 2022.

[40] H.N. Monday, J.P. Li, G.U Nneji,. A.Z. Yutra, B.D. Lemessa, S. Nahar, E.C. James, and A.U. Haq, "The Capability of Wavelet Convolutional Neural Network for Detecting Cyber Attack of Distributed Denial of Service in Smart Grid," 18th International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP 2021, 2021.

[41] Y.Wang, Z. Zhang, J. Ma, Q. Jin, "KFRNN: An Effective False Data Injection Attack Detection in Smart Grid Based on Kalman Filter and Recurrent Neural Network," IEEE Internet of Things Journal, 2022.

[42] Y. Wang, W. Shi, Q Jin, and J. Ma, "An accurate false data detection in smart grid based on residual recurrent neural network and adaptive threshold," IEEE International Conference on Energy Internet, ICEI 2019, 2019.

[43] Y. Wang, D. Chen, C. Zhang, X. Chen, B. Huang, and Cheng, X., "Wide and Recurrent Neural Networks for Detection of False Data Injection in Smart Grids," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2019.

[44] A. Ayad, H.E.Z. Farag, A.Youssef, and E.F. El-Saadany, "Detection of false data injection attacks in smart grids using Recurrent Neural Networks.", 2018 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference, ISGT 2018, 2018.

[45] L. Yang, Y. Zhai, Z. Li, "Deep learning for online AC False Data Injection Attack detection in smart grids: An approach using LSTM-Autoencoder," Journal of Network and Computer Applications, 2021.

[46] M. Alazab, S. Khan, S.S.R. Krishnan, Q.V. Pham, M.P.K. Reddy, and T.R. Gadekallu, "A Multidirectional LSTM Model for Predicting the Stability of a Smart Grid," IEEE Access, 2020.

[47] N.C. Enriquez, and Y. Weng, "Attack Power System State Estimation by Implicitly Learning the Underlying Models, " IEEE Transactions On Smart Grid, VOL. 14, NO. 1, January 2023.

[48] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," IEEE Trans. Power Syst., vol. 26, no. 1, pp. 12–19, Feb. 2011.G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

.