**RESEARCH PAPER – 1: Sufficiency of Ensemble Machine Learning Methods for Phishing Websites Detection**

## CHAPTER - 1 INTRODUCTION

With the expansion of the Internet and the ubiquity of social media, data breaches have consequently emerged as one of the main concerns in cyber security fields. Most security problems and data breaches are usually caused by malicious criminals. Phishing is a common form of cybercrime when hackers attempt to lure individuals into divulging private information, such as bank account details, credit card number, and even employee login credentials for use in unauthorized access to a specific company. To lure a victim, hackers create fraudulent messages that seem to come from a trustworthy person or entity but actually contain disguised links. Then, they send these fake messages to the targets by email or instant messages. If the victim is tricked by the malicious link, confidential data of him or her will be stolen in this cyber fraud. Since the coronavirus pandemic, people are ordered to work remotely, Covid-19-themed phishing attacks have spiked. Phishers take advantage of the virus-related fear and anxiety of the public in the wake of the spread of the virus. Emails allegedly providing ways to stop the coronavirus outbreak were the most common kind of phishing emails employed. In order to boost the likelihood of success, phishing attempts that occurred during the pandemic also had distinctive features, for instance, the registration of covid-related domains soared during the first months of the pandemic. Threats on social media continued to escalate, with a 47% increase from Q1 to Q2 2022, according to a recent trends report by the APWG (Anti-Phishing Working Group). Artificial Intelligence (AI) is an emerging science, which has captured tremendous attention over the past decades. It investigates how to build intelligent machines that can creatively find solutions to problems without human intervention. Machine Learning (ML) is a branch of AI that gives machines the capability to automatically learn and make decisions from experience. As a subset of ML, deep learning (DL) employs neural networks with a structure resembling the human neural system to analyze a wide range of variables. Researchers in the cyber security domain have con- ducted various AI solutions to detect illegal phishing attack.

## CHAPTER - 2 LITERATURE REVIEW

Based on the methodologies used, phishing detection solutions can be categorized into many different groups including blacklist and whitelist, heuristic-based method, visual similarity, machine learning, deep learning, and hybrid. This section mainly talks about two categories: ML-based phishing detection techniques and DL-based phishing detection approaches in the literature.

### 2.1 ML-BASED PHISHING DETECTION

Phishing detection has become a critical area of focus in cyber security, with machine learning (ML) methods playing a central role in identifying malicious attempts. Among the various ML approaches supervised, semi-supervised, unsupervised, and reinforcement learning supervised learning is the most widely used due to its ability to learn patterns from labeled datasets. These systems typically extract features such as URLs, hyperlinks, page content, and hybrid combinations to differentiate phishing from legitimate websites. The effectiveness of such models is highly dependent on the quality and relevance of the selected features, as well as the underlying algorithm. Techniques like Support Vector Machines (SVM), Random Forests, AdaBoost, and LightGBM have shown promising results. Butnaru et al. proposed a model using URL features that outperformed Google Safe Browsing and remained effective over time, though concerns about adversarial attacks remain. Jain and Gupta's approach, based on hyperlink analysis within HTML code, introduced six new features that improved performance and response time, but the model is limited if attackers modify the source code entirely. To tackle the challenge of selecting relevant features, Chiew et al. developed the Hybrid Ensemble Feature Selection (HEFS) framework, which uses a gradient-based method to significantly reduce feature size while maintaining 94.6% detection accuracy. Building on this, a newer framework introduced the use of feature importance techniques Mean Decrease in Impurity (MDI), Permutation, and SHAP to dynamically rank and select optimal features. This method not only surpasses HEFS in detection accuracy (96.83%) but also automates the process, making it more robust, flexible, and applicable across different datasets. Such advancements highlight the ongoing evolution and importance of intelligent feature engineering in creating effective phishing detection systems.

### 2.2 DL-BASED PHISHING DETECTION

Deep Learning (DL) has become a promising alternative to traditional Machine Learning (ML) in phishing detection due to its strong ability to extract hidden patterns from complex and large datasets. Various DL techniques such as Multi-Layer Perceptron (MLP), Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and hybrid models have been widely adopted. Yerima and Alzaylaee developed a CNN-based

model with high detection accuracy, using a structured 1D-CNN architecture that outperformed several traditional ML models. However, tuning DL models often involves a time-consuming process of adjusting multiple hyper parameters. Similarly, Li et al. introduced an LSTM-based system for detecting phishing emails in large datasets, combining KNN and K-Means in a sample expansion stage to overcome the challenge of limited labeled data. Their approach achieved an accuracy of about 95%. While DL excels at handling large-scale data and delivers high performance, it faces significant challenges, including the need for extensive labeled datasets, high computational requirements, and a lack of interpretability. As noted in recent reviews, each DL model is suited to specific data types CNNs for image-like data and RNNs for sequential data like text but the complexity and opacity of these models can limit their practical usability and understanding. In a recent comprehensive DL-based review in the phishing detection field, Do et al. indicated that Each DL algorithm has unique properties that make it ideal for a specific application. For example, RNN is more appropriate for processing sequential data such as natural language and text. When analyzing two dimensional data, such as images and videos, CNN produces better results. In addition, the main drawback is that supervised DL requires a massive amount of labeled instances, which adds a high level of computational complexity to the detection system. Additionally, DL models are unable to justify the inference they draw. It would be tough to comprehend the relationship between input attributes and output decisions.

## CHAPTER - 3 DATASET AND FEATURES

Several high-quality phishing datasets are widely used by various authors in their research, such as UCI_2015, Mendeley_2018, and Mendeley_2020. Phishing instances are usually derived from PhishTank, which is a cooperative repository for data and information about phishing attacks on the Internet. Other legitimate instances are from Alexa, DMOZ, and Common Crawl. Features used in phishing detection are usually extracted from URLs (protocol, domain, path) and other external resources. In this section, it will give an introduction and comparison of these three popular phishing datasets.

### A. UCI_2015

University California Irvine Machine Learning Repository (UCI) is a common repository that contains both fraudulent and trustworthy website URLs, which is popular among phishing detection researchers. The dataset was donated in 2015 and collected primarily from PhishTank and MillerSmiles archives. Although the UCI dataset is widely used, it is now too old to be used for modern phishing detection algorithms development.

### B. MENDELEY_2018

48 features are contained in the dataset Mendeley_2018, which includes 5000 malicious and 5000 legitimate instances. The legal websites are derived from Alexa and common crawl, whereas phishing instances are from PhishTank and OpenPhish. Based on this dataset, L. Chiew et al. proposed the HEFS framework mentioned. It shows a list of features in Mendeley_2018.

### C. MENDELEY_2020

Dataset Mendeley_2020 is the primary dataset utilized in our research, which consists of two sub-datasets: dataset_full and dataset_small. There are 88647 instances in the full dataset and 58645 instances in the small dataset. Data were collected from PhishTank and Alexa ranking. This dataset contains features, for better understanding, it redivided them into 8 groups.

### D. COMPARISON

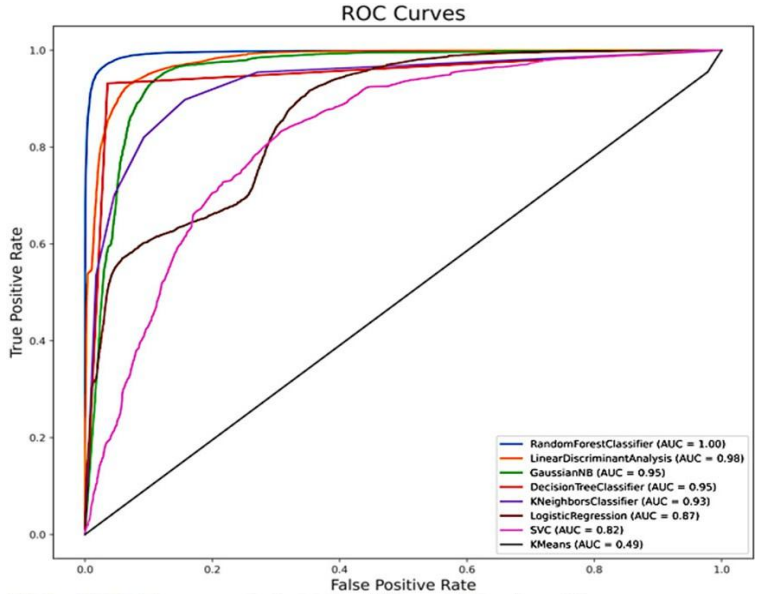
In addition, all features in dataset UCI_2015 were transformed into Boolean type based on specified rules, making it difficult for further analysis. Dataset Mendeley_2020 was selected in our research for its quantity in instances and features.

## CHAPTER - 4 ML-BASED PHISHING DETECTION RESULTS

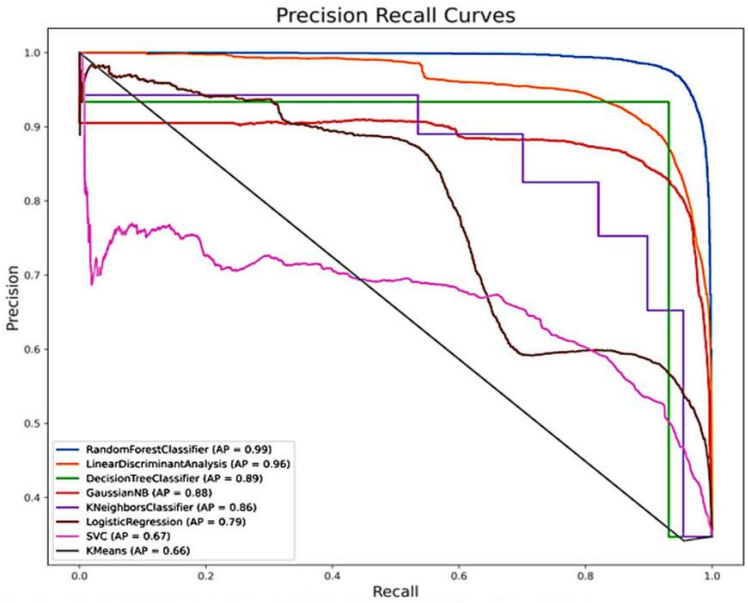
In this section, this paper performed an empirical analysis of various traditional ML algorithms for phishing detection. First, traditional ML algorithms including K-Means Clustering (KMeans), Support Vector Machine (SVM), Naive Bayes Classifier (NB), K-Nearest Neighbor (KNN), Logistic Regression (LR), Linear Discriminant Analysis (LDA), Classification and Regression Tree (CART), and Random Forest (RF) were utilized to classify. Then, results by using ensemble ML methods including RF, AdaBoost, GBDT, XGBoost, and LightGBM were compared in the second sub-section. The same as most studies, performance was analyzed using Accuracy, Precision, Recall, F1 score, ROC Curve, and P-R Curve.

## 4.1 TRADITIONAL ML ALGORITHMS

On Jupyter Notebook (6.4.3), all of the models were trained using the scikit-learn (1.1.2) library with Python (3.8.11) programming language. it used 10-fold cross-validation in our studies on the full dataset in Mendeley_2020. As a result, RF shows the best performance on all metrics with a 97.01% accuracy rate. As can be seen from the graphs, the highest value of Area Under Curve (AUC) belongs to RF, which means that it can separate the positive class and negative class correctly. Besides, RF presents the ability to return accurate results (high precision), as well as high positive results (high recall) at the same time in P-R Curves.




ROC curves of eight traditional ML classifiers.



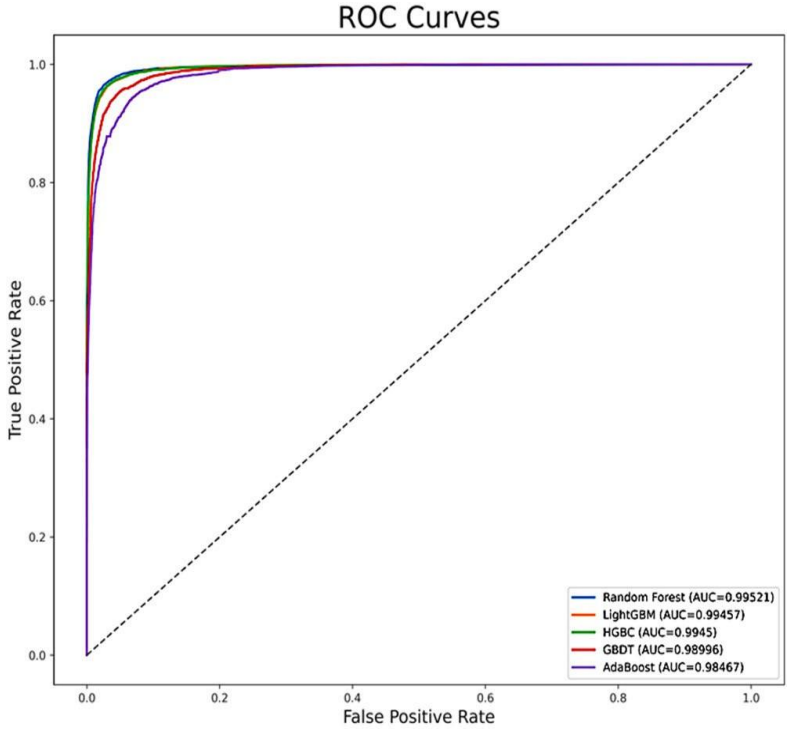

P-R curves of eight traditional ML classifiers.

## 4.2 ENSEMBLE ML ALGORITHMS

The learning algorithms known as ‘‘ensemble ML methods’’ classify new data by performing a (weighted) vote on the predictions made by each classifier. They are considered as the state-of-the-art solutions for many ML challenges. This paper implemented 5 ensemble ML methods on the dataset including AdaBoost, Gradient Boosted Decision Trees (GBDT), LightGBM (version 3.3.3), Histogram-Based Gradient Boosting (HGB), and the most popular ensemble method Random Forest (RF). In this experiment, This paper split the original dataset into two parts, using 70% for training and 30% for testing. LightGBM shows its high efficiency with minimum training and testing time consumption. It can conclude that ensemble ML methods, in particular the boosting methods, tend to achieve the best performance in phishing classification.

| No | Classifier | Accuracy(%) | Precision(%) | Recall(%) | F1score(%) |
|---|---|---|---|---|---|
| 1 | KMeans | 62.60 | 51.67 | 13.78 | 16.96 |
| 2 | SVM | 75.46 | 67.30 | 55.89 | 61.06 |
| 3 | NB | 83.85 | 87.98 | 61.48 | 72.37 |
| 4 | KNN | 86.95 | 81.72 | 80.00 | 80.85 |
| 5 | LR | 89.76 | 87.38 | 82.27 | 84.59 |
| 6 | LDA | 91.54 | 82.77 | 95.26 | 88.58 |
| 7 | CART | 95.16 | 93.01 | 92.88 | 92.98 |
| 8 | RF | 97.01 | 95.44 | 95.93 | 95.69 |

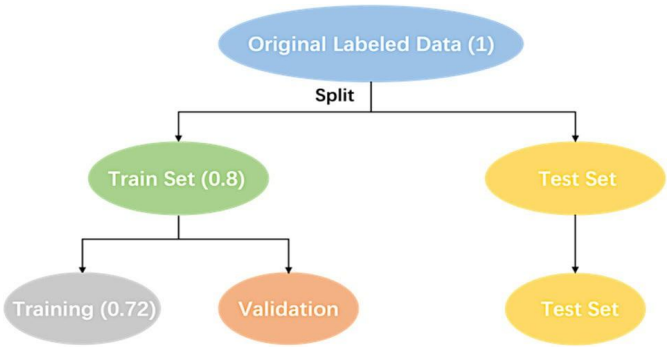
Performance metrics of various traditional ML algorithms.




ROC curves of five ensemble ML classifiers.

## CHAPTER - 5 DL-BASED PHISHING DETECTION RESULTS

The goal of this section is to assess the performance of current popular DL-based methods including FCNN, LSTM, and CNN. Fully Connected Neural Networks (FCNN) are constituted by a sequence of completely connected layers that have the primary advantage of being ‘‘structure agnostic,’’ meaning that no special assumptions about the input are required. LSTM is a particularly unique type of Recurrent Neural Network (RNN) that performs significantly better than the normal version. It was introduced by Hochreiter and Schmidhuber and several researchers have since improved and popularized it. LSTMs are specifically designed to prevent the long-term dependency problem. CNN is renowned for its ability to recognize simple patterns in a multi-dimensional task, and as a result, it has had success processing 2D signals like images and video frames. However, a 1D CNN model can also be used to process datasets with a one-dimensional structure. In the following subsections, the experiment setup and data division are described, following the result and comparison.

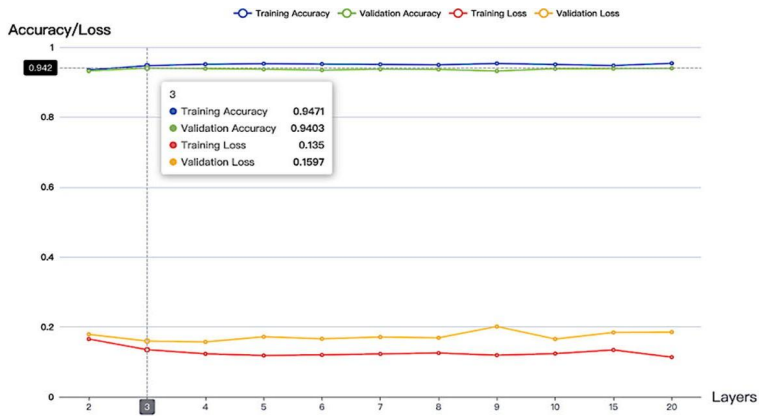



Dataset is divided into three parts.

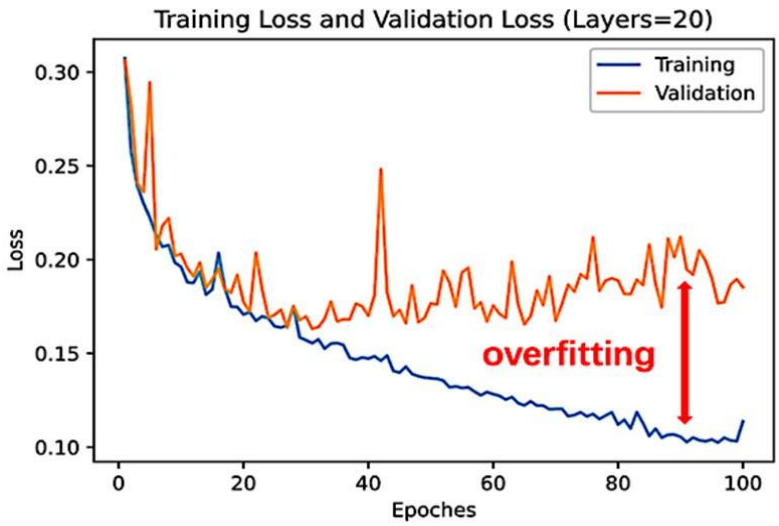
### 5.1 EXPERIMENTAL SETUP

This paper built three DL-based models by using Python with Tensorflow and Keras library on Jupyter Notebook. The dataset was divided into three parts: training dataset, validation dataset, and test dataset. The train dataset is 80% of the original dataset, and 20% is the test dataset. Furthermore, 10% of the train dataset is used as a validation dataset shown in figure.

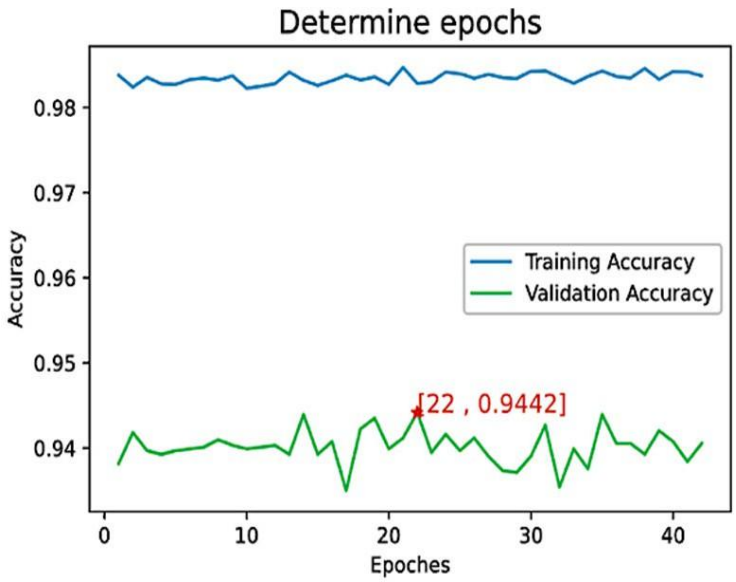
Fully connected layers are usually used for classification, in order to build the FCNN model, it is essential to decide the number of layers, it set different layers to observe the changes in accuracy and loss on the validation dataset. When the number of layers rises, the accuracy rate and loss are basically flat, and the validation accuracy rate is at its highest (0.9403) when the number of layers is 3. Overfitting occurs when the number of layers is 20, which indicates that the model fits perfectly against its training data but fails to perform accurately against the unseen (test) dataset, violating its purpose.



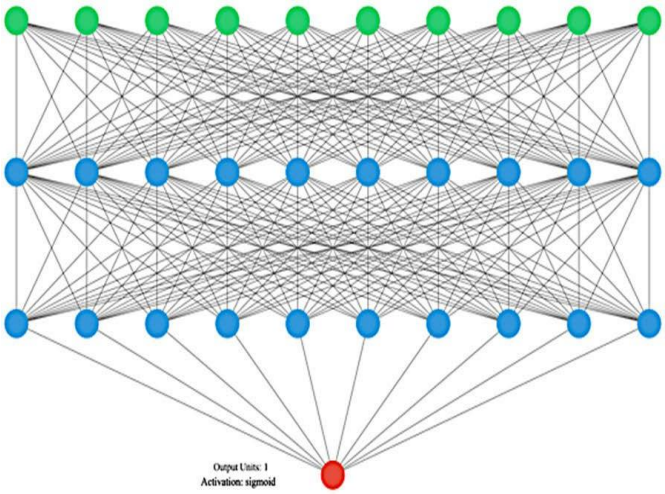

Accuracy and loss vs. number of layers in FCNN.




Overfitting occurs in the 20-layers FCNN model.




Accuracy vs. epochs in the 3-layers FCNN model.

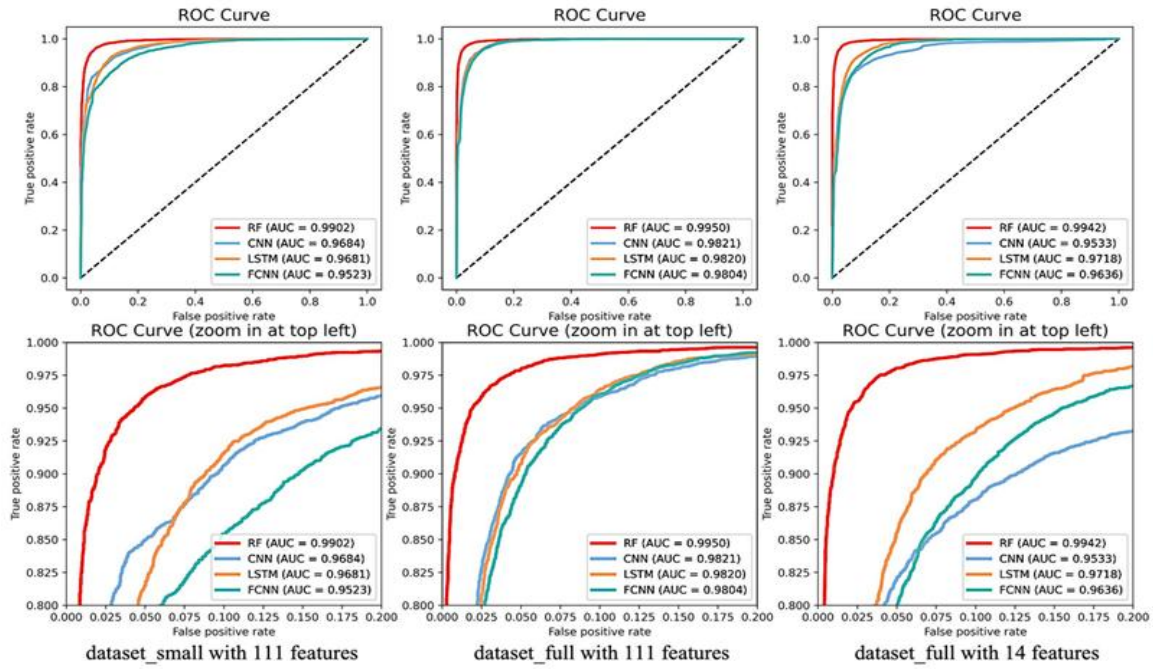



Our 3-layers FCNN model.

Procedure can be seen as a basic example of parameter settings in DL-based methods. Parameters can differ between different DL models, such as the number of layers in the model, batch size, the number of epochs, type of optimizer, type of activation function in hidden layers and output layer, etc. Based on these steps, it built a 3-layers LSTM model with one dropout layer and one dense layer. In addition, a 6-layers CNN model was constructed in the research. Table 7 lists the parameter settings for these DL architectures.

## 5.2 RESULT AND COMPARISON

To increase the reliability of classifications, models include RF were tested on three datasets: dataset_small with 111 features, dataset_full with 111 features, and dataset_full with 14 selected features in our previous work. Evaluation metrics consist of training time consumption, precision, recall, AUC, and accuracy. From the table, it observed the following phenomenon that needs to be emphasized. First, all the classifiers perform better when data is getting bigger from dataset_small to dataset_full, which indicates that significant datasets are typically necessary for AI to reach high accuracy. Second, it is surprising that RF out-performs other DL models with the highest testing accuracy rate 96.94%, whereas that of CNN, FCNN, and LSTM are 91.38% 90.13%, and 89.73%, respectively. This result casts a new light on the performance of RF model. Third, RF model has the lowest training time, which is sensible because the computation complexity of DL-based models is always high. Note that it only record the training time cost of its best fine- tuning state for each individual model. Furthermore, it also conducted an experiment to compare the performances of the selected features against full features on dataset_full. Results showed that RF only experiences a minimal accuracy deterioration of 0.1% (96.94% to 96.84%) while achieving a massive reduction in the dataset. Compared to RF, DL models suffer from serious decreases in testing accuracy rate with selected features. It also presents ROC Curves of the 4 classifiers, where lower plots are larger versions zooming in at the top left. The curves and Area Under the ROC Curve (AUC) values offer a more comprehensive insight into the performances of the models. In every graph, RF clearly shows incomparable curves against other DL models. As a result, the evaluation results have validated that RF is advantageous and highly effective when working with selected features and real-time applications in distinguishing between legitimate and phishing websites. The implications of these findings are discussed in the following Section to highlight the sufficiency of ensemble ML methods in phishing detection and navigate the future directions.




ROC curves of RF, CNN, LSTM, and FCNN on three different datasets.

## CHAPTER - 6 DISCUSSION AND CONCLUSION

Previous sections have compared classification performances of various ML models and DL models. In this Section, it discuss the advantages and disadvantages between the two groups and draw our conclusion.

Deep Learning is considered to be the state-of-the-art solution to various problems with the advantages of dealing with big data and generating features automatically over Machine Learning. However, model architecture design, manual parameter tuning, high training time costs, computational complexity, and deficient accuracy performance are the most prevalent problems with DL approaches, as discussed.

Ensemble ML techniques represented by RF are usually regarded as a crystallization of wisdom of various ML methods. In ensemble methods, by combining different models, the risk of selecting an improper decision is reduced, and thus, the forecast performance is improved. In our experiments, CART, RF, and Boosting methods obtain better performances in phishing classification. This is potentially due to these ensemble methods benefit from the dynamic changing of assigned weight to each instance in the iteration process, making it more robust and stable than traditional ML algorithms. For instance, AdaBoost's basic principle is to concentrate on cases that were previously incorrectly classified when training a new inducer. In the initial iteration, each instance is given the same weight, after which the weights of incorrectly categorized instances increase and those of correctly identified examples decrease. Additionally, based on their total prediction performances, the individual basic learners are also given voting weights. Hence, ensemble ML methods decrease both bias and variance of variable techniques while increasing the variance for stable classifiers, making them more suitable for classification tasks.

As a typical binary classification problem, ML-based phishing detection solutions are questioned on the ability to handle big data and extract features. Researchers believe that the process of feature selection relies on professional knowledge and reduplicative experiments, which is considered to be tedious, labor-intensive, and susceptible to human mistakes. However, this problem can be effectively and efficiently resolved by utilizing automatic feature selection methods, for example, our feature selection framework achieves a remarkable 87.6% reduction in feature quantity with suffering from only a 0.1% deterioration in detecting accuracy, making it possible for up-date training and real- time detecting in a production environment. In another hand, phishers are also employing the latest schemes to execute attacks, phishing features are under evolution constantly. The phishing websites features cannot be generated once and for all,

conversely, it should be a continuous updating and accumulating process, in which researchers are supposed to pay efforts.

To sum up, our experiments and discussions offer a significant insight into the sufficiency of ensemble ML methods for anti-phishing techniques. As for future work, it will validate our conclusion on various datasets with more features and more instances. In addition, further efforts need to be taken to avoid the inefficiency when detecting zero-day attacks. This plan to extract features of the latest phishing websites and train our ensemble ML method at intervals. Then, by observing the variation trends in newly evolving phishing patterns, it would like to find a balanced renewal frequency for extracting features and training models to maintain high detection accuracy. Last but not least, as a practical tool, a phishing detection architecture is supposed to be deployed in a real-world production environment (e.g. web browser) to verify its effectiveness against phishing attacks eventually.

**Research Paper – 2: A Machine Learning Framework for Early-Stage Detection of Autism Spectrum Disorders**

## CHAPTER - 1 INTRODUCTION

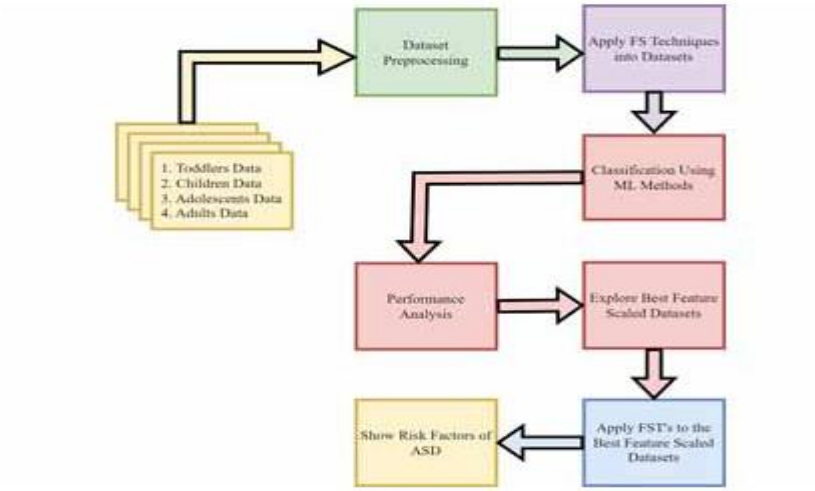
Autism Spectrum Disorder (ASD) is a neurodevelopmental condition affecting early brain development, leading to challenges in communication, social interaction, and behavior. Early diagnosis and intervention are crucial for improving developmental outcomes, but traditional behavioral diagnostic approaches are complex and often delayed. Recently, mobile apps and open-access datasets (e.g., from UCI and Kaggle) have been developed to aid early ASD detection across different age groups.

Numerous studies have explored machine learning (ML) for ASD diagnosis, using models like Random Forest, Decision Tree, SVM, Logistic Regression, and Deep Neural Networks. Some have also applied feature selection and optimization techniques to enhance accuracy and interpretability. Despite these advances, limitations remain in feature scaling strategies, model validation, and comprehensive analysis across all age categories.

In this study, this paper present a robust ML-based framework for ASD detection using four benchmark datasets: Toddlers, Children, Adolescents, and Adults. It apply four modern Feature Scaling (FS) methods Quantile Transformer, Power Transformer, Normalizer, and Max Abs Scaler to preprocess the data. These scaled datasets are classified using eight effective ML algorithms: AdaBoost, Random Forest, Decision Tree, KNN, Gaussian Naïve Bayes, Logistic Regression, SVM, and LDA.

Additionally, this paper implement four Feature Selection Techniques (FSTs) Info Gain, Gain Ratio, ReliefF, and Correlation Evaluator to identify the most significant ASD risk factors. This framework emphasizes the impact of FS methods and FST tuning on classification performance, helping select optimal features and models for each age group.

Unlike earlier works, our approach uses modern FS techniques, more comprehensive FST tuning, and a broader range of up-to-date ML classifiers. The proposed model shows improved accuracy and interpretability, offering valuable insights for ASD diagnosis and aiding clinicians in early detection.




Sequential workflow for early-stage detection

To this end, the key contributions of this paper are summarized as follows.

- It develop a generalized ML framework for early-stage detection of ASD in people of different ages.
- It solve the imbalanced class distribution issue through Random Over Sampler to avoid the ML models being biased towards the majority class samples.
- It select the best Feature Scaling (FS) method to map individual ASD dataset's feature values to improve the prediction performance.
- It investigate eight simple but effective ML approaches on each feature-scaled ASD dataset, analyze their classification performances and identify the best FS techniques for each ASD dataset.
- Furthermore, this also calculate and analyze the feature importance values on each best feature-scaled ASD dataset based on four FSTs to identify the risk factors for ASD prediction.
- Finally, it also perform extensive experiments and comparisons using four different standard ASD datasets.

# CHAPTER - 2 MATERIALS & METHODS

## 2.1 DATASET DESCRIPTION

It collect the four ASD datasets (Toddlers, Adolescents, Children, and Adults) from the publicly available repositories: Kaggle and UCI ML. The authors in created the ASDTests smartphone app for Toddlers, Children, Adolescents, and Adults ASD screening using QCHAT-10 and AQ-10. The application computes a score of 0 to 10 for every individual, with which the final score is 6 out of 10 which indicates an individual has positive ASD. In addition, ASD data is obtained from the ASDTests app while open-source databases are developed in order to facilitate research in this area. The detailed description of the Toddlers, Children, Adolescents, and Adults ASD datasets.

## 2.2 METHOD OVERVIEW

This research aims to create an effective prediction model using different types of ML methods to detect autism in people of different ages. First of all, the datasets are collected, and then the preprocessing is accomplished via the missing values imputation, feature encoding, and oversampling. The Mean Value Imputation (MVI) method is used to impute the missing values of the dataset. Then, the categorical feature values are converted to their equivalent numerical values using the One Hot Encoding (OHE) technique. As such, a Random Over Sampler strategy is used to alleviate this issue. After completing the initial preprocessing, the dataset's feature values are scaled using four different FS techniques i.e., QT, PT, Normalizer, and MAS. The feature-scaled datasets are then classified using eight different ML classification techniques i.e., AB, RF, DT, KNN, GNB, LR, SVM, and LDA. Comparing the classification outcomes of the classifiers on different feature-scaled ASD datasets, the best-performing classification methods, and the best FS techniques for each ASD dataset are identified. After those analyses, the ASD risk factors are calculated, and the most important attributes are ranked according to their importance values using four different FSTs i.e., IGAE, GRAE, RFAE, and CAE. To this end, Proposed research pipeline to analyze the ASD datasets and calculate the risk factors that are most responsible for ASD detection.

## 2.3 MACHINE LEARNING METHOD

### 2.3.1 ADA BOOST (AB)

AB is a tree-based ensemble classifier that incorporates many weak classifiers to reduce misclassification errors. It selects the training set and iteratively assigns the weights depending on the previous training precision for retraining the algorithm. In order to train any weak classifier, an arbitrary subset of the full training set is used and AB assigns weights to each instance and classifier. The following equation defines the combination of several weak classifiers:

$$H(x) = \text{Sign}(\Sigma\, \alpha t\, ht(x))$$

where H(x) defines the output of the final model through combining the weak classifiers and ht(x) represents the output of classifier t for input x and t specifies the weight assigned to the classifier. αt is calculated as follows.

$$\alpha t = 0.5 * \ln(1 - E)/E$$

where E denote the error rate. The following equation is utilized to update the weights of each training sample-label pair (xi, yi).

### 2.3.2 RANDOM FOREST (RF)

RF is a decision tree-based ensemble classification method and follows the split and conquer technique in the input dataset to create multiple decision-making trees (known as the forest). It works in two phases. At first, it creates a forest by combining the 'N' number of decision trees and in the second phase, it makes predictions for each tree generated in the first phase. The working process of the RF algorithm is
illustrated below:

1) Select random samples from the training dataset.
2) Construct decision trees for each training sample.
3) Select the value of 'N ' to define the number of decision trees.
4) Repeat Steps 1 and 2.
5) For each test sample, find the predictions of each decision tree, and assign the test sample a class value based on majority voting.

### 2.3.3 DECISION TREE (DT)

DT follows a top-down approach to build a predictive model for class values using training data-inducing decision making rules. This research utilized the information gain method to select the best attribute. Assuming Pi, the probability such that xi D, exists to a class Ci, and is predicted by, |Ci, D|/| D|.

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

where Info(D ) is the average amount of information needed to identify Ci of an instance, xi D and the objective of DT is to divide repeatedly, D, into sub datasets D1, D2....Dn. The following equation estimates the Info A(D):

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

Finally, the following equation calculates the information gain value

$$Gain(A) = Info(D) - Info_A(D)$$

### 2.3.4 K-NEAREST NEIGHBORS (KNN)

KNN classifies the test data by utilizing the training data directly by calculating the K value, indicating the number of KNN. For each instance, it computes the distance between all the training instances and sorts the distance. Further more, a majority voting technique is employed to assign the final class label to the test data. This research applies Euclidean distance to calculate the distances among instances. The following equation represents the Euclidean distance calculation

$$D_e = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2}$$

where De indicates the euclidean distance, Xi denotes the testing sample values, Yi specifies the training sample values and n represents the total number of sample values.

### 2.3.5 LOGISTIC REGRESSION (LR)

Based on a given dataset of independent variables, logistic regression calculates the likelihood that an event will occur, such as voting or not voting. Given that the result is a probability, the dependent variable's range is 0 to 1. In logis- tic regression, the odds that is, the likelihood of success divided by the probability of failure-are transformed using the logistic formula. The following formulae are used to express this logistic function, which is sometimes referred to as the log odds or the natural logarithm of odds:

$$p = \frac{1}{1 + e^{-x'}}$$

where p denotes the probability of instance x. At the time of model training, for each instance x1, x2, x3....xn the logistic coefficients will be b0, b1, b2....bn. The stochastic gradient descent method estimates and updates the values of the coefficients.

### 2.3.6 GAUSSIAN NAIVE BAYES(GNB)

GNB algorithm follows a normal distribution and is used for classification when all the data values of a dataset are numeric. To compute the probability values of any instance with respect to the class value mean and standard deviation are calculated for each attribute of the dataset. Consequently, for testing, when any instance comes, it utilizes the mean and standard deviation values to calculate the probability of the test instance. The necessary equations are given below:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\delta = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2$$

$$f(x) = \frac{1}{\sqrt{2\pi}} * \frac{1}{\delta} * e^{-(x-\mu)2}$$

where μ indicates the mean, δ represents standard deviation, xi denotes all samples in a particular column, n indicates the total number of samples and fx presents the conditional probability of class value.

### 2.3.7 SUPPORT VECTOR MACHINE (SVM)

SVM is used to classify both linear and non-linear data and mostly works well for high-dimensional data with non- linear mapping. It explores the decision boundary or optimal hyperplane to separate one class from another. This study used Radial Basis Function (RBF) as a kernel function and SVM automatically defines centers, weights, and thresholds and reduces an upper bound of expected test error.

The following equation represents the RBF function:

$$K(x, x') = exp(-\frac{(||x - x'||)^2}{2\delta^2})$$

### 2.3.8 LINEAR DISCRIMINANT ANALYSIS (LDA)

LDA is a dimensionality reduction technique but can be used for classification by exploring the linear combination of features. LDA uses the Bayes theorem to estimate the probability. Let us, consider k classes and n training samples that are defined as {x1, x2….xn} with classes zi‹{1….k}. The prior probability is assumed to display as Gaussian distribution φ(x|μk , Σ) in each class. The model estimation is defined as follows:

$$a_k = \frac{\sum_{i=1}^{n} 1 * (z_i = k)}{n}$$
$$\mu_k = \frac{\sum_{i=1}^{n} x_i * 1 * (z_i = k)}{\sum_{i=1}^{n} 1 * (z_i = k)}$$
$$\Sigma = \frac{\sum_{i=1}^{n} (x_i - \mu_{zi})(x_i - \mu_{zi})^T}{n},$$

where ak denotes the prior probability, μk defines mean of all classes, Σ indicates the sample covariance of the class means.

# CHAPTER - 3 EXPERIMENTAL RESULTS ANALYSIS

## 3.1 EXPERIMENTALSETUP

In order to conduct the experiment, an open-source cloud- based service named Google Collaboratory provided by Google is utilized. The scikit-learn package of Python programming language is used to complete the data preprocessing, feature scaling, feature selection, and classification tasks. In this work, a 10-fold cross-validation technique is utilized to construct prediction models using four different ASD (Toddlers, Children, Adolescents, and Adults) datasets. In 10-fold cross-validation, during training, the datasets are randomly divided into equal 10 folds. During model building, 9 folds are used and training and the remaining one is used for testing. Hence, this procedure is repeated 10 times, and finally, average the results. Here, due to the lack of enough samples in the datasets, 10-fold cross- validation is used to prevent the model from overfitting and reducing the variance during model building and generalize the model with a small amount of data. If it perform hold- out validation with a fixed test set, then, there would have a possibility of potential overfitting during model building and it will increase the variance and thus cannot generalize the prediction model for unseen test data. Various statistical evaluation measures including accuracy, Receiver Operating Characteristics (ROC) curve, F1-score, precision, recall, Mathews Correlation Coefficient (MCC), Kappa score, and Log loss are considered to justify the experimental outcomes. The evaluation measures are calculated using the following formulae.

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP}$$

$$Precision = \frac{TP}{FP + TP}$$

$$Recall = \frac{TP}{FN + TP}$$

$$F1 - Score = \frac{2TP}{FN + FP + 2TP}$$

The following terms represent the above equations. TP = True Positive; TN = True Negative; FP = False Positive; FN = False Negative; po is the relative observed agreement among raters; and p is the hypothetical probability of chance agreement; y is the actual/true value and yr is the prediction probability of each observation.

## 3.2 ANALYSIS ON ACCURACY

Accuracy represents the actual prediction performance of any classifier. The higher the value of accuracy indicates better prediction and lower the miss-classification. In this case, LDA delivers the best accuracy of 97.12% for the normalizer-scaled Adolescent dataset. Moreover, while investigating the results of the feature-scaled Adult dataset, it is seen that both

the QT and normalizer-scaled datasets perform better than the other FS methods. In both of the cases, LDA achieves the best accuracy value of 99.03%. Additionally, the accuracy values of various ML classifiers on feature-scaled Toddlers, Children, Adolescents, and Adult datasets are contrasted.

### 3.3 ANALYSIS ON PRECISION

Precision represents positive predictive value and a higher value of precision means the true positive value is high and the false positive value is low. The precision values of various classifiers on different feature-scaled datasets are presented Analyzing the precision values of the Toddler dataset, it is found that the AB classifier provides the best precision of 99.95% while PT is used as the FS method. While reviewing the feature-scaled Children dataset, it is noticed that the LR classifier obtains the highest precision of 96.16% for MAS in classifying ASD. Furthermore, inspecting the feature-scaled Adolescent dataset, it observe that DT delivers the best precision of 97.25% while using PT as FS method. Moreover, while investigating the results of the feature-scaled Adult dataset, it is seen that both the QT-transformed datasets perform better than the other FS methods. In that case, SVM achieves the best precision value of 98.16%. Additionally, the precision values of various ML classifiers on feature- scaled Toddlers, Children, Adolescents, and Adult datasets are contrasted.

### 3.4 ANALYSIS ON RECALL

Recall represents a true positive rate and a higher value of recall means the true positive value is high and the false negative value is low. When the true positive is high and the false negative is low that means better predic- tion. The recall values of various ML classifiers on different feature-scaled datasets. While reviewing the recall results of the feature scaled Toddler dataset, it is observed that AB obtains the highest recall of 98.45% for the normalizer scaled Toddler dataset. Investigating the feature-scaled Children datasets, it find that LR delivers the best recall value of 97.72% while nor- malizer as FS method. Moreover, inspecting the recall results of feature-scaled adolescent datasets, it is noticed that AB achieves the highest recall of 97.36% for normalizer-scaled Adolescent datasets. Finally, it analyze the outcomes of feature-scaled Adult datasets and find that RF, KNN, and LR deliver the highest recall of 100.00% for PT, while DT and KNN obtain the best recall of 100.00% for PT and KNN, LR also obtains 100.00% recall value for MAS-scaled adult’s datasets. Besides, it also compare the recall values of various ML classifiers on feature-scaled Toddlers, Children, Adolescents, and Adult datasets.

### 3.5 ANALYSIS ON ROC

The ROC value indicates the ability of any classifier to distin- guish between positive and negative classes. The ROC values of various ML classifiers on different feature-scaled datasets are presented in Table 8. While reviewing the ROC results of the feature-scaled Toddler dataset, it is observed that LR obtains the highest ROC of 99.99% for both QT and PT and AB achieves 99.99% for the normalizer method. Investigating the feature-scaled Children dataset, it is found that GNB delivers the best ROC value of 99.73% using normalizer as the FS method. Moreover, inspecting the ROC results of the feature-scaled Adolescent dataset, it notice that both AB and LDA achieve the highest ROC of 99.72% for QT and MAS-scaled datasets. Finally, it analyze the outcomes of feature-scaled Adult datasets and find that LDA delivers the highest ROC value of 99.99% while using PT and normalizer as the FS methods. It compare the ROC values of various ML classifiers on feature-scaled Toddlers, Children, Adoles- cents, and Adult datasets.

### 3.6 ANALYSIS ON THE F1-SCORE

F1-score takes the harmonic mean of the precision and recall values and a higher value of it indicates better prediction. The F1-score values of various ML classifiers on different feature scaled datasets are presented in Table 9. While reviewing the F1-score results of the feature scaled Toddler dataset, it observe that AB obtains the highest F1-score of 99.14% for the normalizer scaled Toddler dataset. Investigating the feature-scaled Children dataset, it is found that AB delivers the best F1-score value of 97.02% while using QT and normalizer as FS methods. Moreover, inspecting the F1-score results of feature-scaled Adolescent datasets, it notice that AB achieves the highest F1-score of 97.69% for the QT-scaled Adolescent dataset. Finally, it analyze the outcomes of the feature-scaled Adult dataset and notice

### 3.7 ANALYSIS ON KAPPA

Kappa score measures the degree of agreement between true class and predicted class. The higher value of kappa means a better prediction which indicates a higher degree of agreement between actual and predicted values. The kappa values of various ML classifiers on different feature-scaled datasets are presented in Table 10. While reviewing the kappa results of the feature-scaled Toddler dataset, it is observed that both the normalizer and MAS-scaled datasets provide the best kappa value and outperform the other FS meth- ods. Consequently, both LR and LDA obtain the highest MCC of 99.31% for normalizer and MAS-scaled Toddler datasets. Investigating the feature-scaled Children datasets, it is found that AB delivers the best

kappa value of 93.78% using normalizer as FS method. Moreover, inspecting the kappa results of feature-scaled Adolescent datasets, it notice that LDA achieves the highest MCC of 94.02% for both QT and PT-scaled datasets respectively. Finally, it analyze the outcomes of feature-scaled Adult datasets and see that both LR and LDA deliver the highest kappa value of 99.02% while using QT and normalizer as the feature scaling methods. Besides, it also compare the kappa values of various ML classifiers on feature-scaled Toddlers, Children, Adolescents, and Adult datasets.

### 3.8 ANALYSIS ON LOG LOSS

The log loss value indicates how close the prediction probability is to the true values. The lower the log loss value, the better the prediction. The log loss values of various ML classifiers on different feature-scaled datasets. While reviewing the log loss results of the feature-scaled Toddler and children datasets, it observe that AB obtains the lowest log loss of 0.0802% and 0.98% for the normalizer-scaled toddler and QT and PT-scaled chil- dren. Furthemore, it is noticed that LDA achieves the lowest log loss of 1.12% for QT, PT, and MAS-scaled adolescents datasets. Finally, it analyze the outcomes of feature-scaled Adult datasets and see that both LR and LDA deliver the high- est log loss value of 0.16% while using QT and normalizer as the feature scaling methods. Besides, it also compare the log loss values of various ML classifiers on feature-scaled Toddlers, Children, Adolescents, and Adult datasets

### 3.9 ANALYSIS ON MCC

MCC takes all the coefficient of confusion matrix such as TP, TN, FN and FP into consideration to calculate the degree of correlation. The higher value of MCC represents better pre- diction and strong correlation between actual and predicted class. While reviewing the MCC results of the feature-scaled Toddler dataset, it observe that both LR and LDA obtain the highest MCC of 99.31% for normalizer and MAS-scaled Toddler datasets. Investigating the feature-scaled children datasets, it is found that AB delivers the best MCC value of 93.88% using normalizer as the FS method. Moreover, inspecting the MCC results of feature-scaled Adolescent datasets, it notice that LDA achieves the highest MCC of 94.25% for both QT and PT-scaled datasets respectively. Finally, it analyze the outcomes of feature-scaled Adult datasets and find that both LR and LDA deliver the highest MCC value of 99.03% while using QT as the feature scal- ing method. Besi’des, it also compare the MCC values of various ML classifiers on feature-scaled toddlers, children, adolescents, and adults datasets.

## CHAPTER - 4 DISCUSSION AND EXTENDED COMPARISON

In the previous section, it analyzed four different ASD datasets to build prediction models for different stages of people. In order to do this, it applied various FS methods to those ASD datasets and classified them utilizing eight differ- ent simple but effective ML classifiers and also determined how the FS methods affect the classification performance. Furthermore, it also employed four different FSTs to com- pute the importance of the features which are more responsi- ble for ASD prediction. Inspecting the experimental findings, the best performing classifiers model predicted ASD with AB (99.25%), AB (97.95%), LDA (97.12%), LDA (99.03%) accuracy; AB, LR (99.99%), GNB (99.73%), AB, LDA (99.72%), LDA (99.99%) ROC; AB (99.14%), AB (97.02%), AB (97.69%), LDA (99.11%) F1-score; AB (99.95%), LR (96.16%), DT (97.25%), SVM (98.16%) precision; AB (98.45%), LR (97.72%), AB (97.36%), RF, DT, KNN, LR (100%) recall; LR, LDA (99.31%), AB (93.88%), LDA (94.25%), LR, LDA (99.03%) MCC; LR, LDA (99.31%), AB (93.78%), LDA (94.02%), LR, LDA (99.02%) kappa; AB (0.0802%), AB (0.98%), LDA (1.12%), LR, LDA (0.16%) log loss for Toddlers, Children, Adolescents, Adults datasets respectively. After analyzing the experimental outcomes of different classifiers on feature-scaled ASD datasets, it is found that AB for Toddlers and Children, and LDA for Ado- lescents and Adults outperformed the other ML classifiers in terms of classification performance. Besides, the experi- mental outcomes implied that the normalizer FS method for Toddlers, normalizer FS method for Children, QT FS method for Adolescents, and QT FS method for Adults showed better performance. Additionally, it calculated the feature impor- tance using the IGAE, GRAE, RFAE, and CAE FST methods on the normalizer-scaled Toddlers, normalizer-scaled Chil- dren, QT-scaled Adolescents, and QT-scaled Adults to enu- merate the risk factors for ASD prediction. This feature importance analysis helps healthcare practitioners decide the most important features while screen- ing ASD cases.

| Dataset | Reference | Accuracy | ROC | F1 | Precision | Recall | MCC | Kappa | Log Loss |
|---|---|---|---|---|---|---|---|---|---|
| Toddlers | Mousumi *et al.* [1] | 97.82 | 99.70 | 97.80 | - | - | - | 94.87 | - |
| | Proposed Model | 99.25 | 99.99 | 99.14 | 99.89 | 98.45 | 98.99 | 98.97 | 0.0802 |
| Children | Omar *et al.* [18] | 92.26 | - | - | - | - | - | - | - |
| | Thabtah *et al.* [17] | 97.80 | - | - | - | 98.00 | - | - | - |
| | Talabani *et al.* [49] | 92.26 | - | - | 88.09 | 96.52 | - | - | - |
| | Mousumi *et al.* [1] | 99.61 | 99.60 | 99.60 | - | - | - | 99.21 | - |
| | Haroon *et al.* [30] | 95.5 | - | 96.00 | 97.00 | 98.00 | 90.10 | - | - |
| | Abitha *et al.* [31] | 94.1 | - | - | - | - | - | - | - |
| | Kamna *et al.* [33] | 95.82 | - | - | - | - | - | - | - |
| | Gupta *et al.* [34] | - | - | 94.71 | 92.59 | 97.09 | 89.32 | - | - |
| | Proposed Model | 97.95 | 99.73 | 97.02 | 96.16 | 97.72 | 93.88 | 93.78 | 0.98 |
| Adolescents | Omar *et al.* [18] | 93.78 | - | - | - | - | - | - | - |
| | Thabtah *et al.* [17] | 94.23 | - | - | - | 92.20 | - | - | - |
| | Talabani *et al.* [49] | 93.78 | - | - | 89.85 | 98.4 | - | - | - |
| | Mousumi *et al.* [1] | 95.87 | 99.00 | 95.90 | - | - | - | 91.74 | - |
| | Kamna *et al.* [33] | 95.82 | - | - | - | - | - | - | - |
| | Gupta *et al.* [34] | - | - | 84.21 | 93.25 | 74.15 | 65.53 | - | - |
| | Proposed Model | 97.12 | 99.72 | 97.69 | 97.25 | 97.36 | 94.25 | 94.02 | 1.12 |
| Adults | Omar *et al.* [18] | 97.10 | - | - | - | - | - | - | - |
| | Thabtah *et al.* [17] | 99.85 | - | - | - | 99.90 | - | - | - |
| | Shuvo *et al.* [25] | 95.71 | - | - | - | 85.71 | - | - | - |
| | Talabani *et al.* [49] | 96.91 | - | - | 90.07 | 96.87 | - | - | - |
| | Mousumi *et al.* [1] | 99.82 | 99.80 | 99.90 | - | - | - | 99.59 | - |
| | Abitha *et al.* [31] | 98.00 | - | - | - | - | - | - | - |
| | Kamna *et al.* [33] | 95.82 | - | - | - | - | - | - | - |
| | Gupta *et al.* [34] | - | - | 94.26 | 97.46 | 91.27 | 92.46 | - | - |
| | Proposed Model | 99.03 | 99.99 | 99.11 | 98.16 | 100.00 | 99.03 | 99.02 | 0.16 |

Comparison with other works.

## CHAPTER - 5 CONCLUSION

In this work, it proposed a machine-learning framework for ASD detection in people of different ages (Toddlers, Children, Adolescents, and Adults). This show that predictive models based on ML techniques are useful tools for this task. After completing the initial data processing, those ASD datasets were scaled using four different types of feature scaling (QT, PT, normalizer, MAS) techniques, classified using eight different ML classifiers (AB, RF, DT, KNN, GNB, LR, SVM, LDA). it then analyzed each feature- scaled dataset's classification performance and identified the best-performing FS and classification approaches. It considered different statistical evaluation measures such as accu- racy, ROC, F1-Score, precision, recall, Mathews correlation coefficient (MCC), kappa score, and Log loss to justify the experimental findings. Consequently, our proposed predic- tion models based on ML techniques can be utilized as an alternative or even a helpful tool for physicians to accurately identify ASD cases for people of different ages. Additionally, the feature importance values were calculated to identify the most prominent features for ASD prediction by employing four different FSTs (IGAE, GRAE, RFAE, and CAE). There- fore, the experimental analysis of this research will allow healthcare practitioners to take into account the most impor- tant features while screening ASD cases. The limitation of our research work is that the amount of data was not suffi- cient enough to build a generalized model for people of all stages. In the future, it intend to collect more data related to ASD and construct a more generalized prediction model for people of any age to improve ASD detection and other neuro-developmental disorders.