KARPAGAM
COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

# Email Phishing Detection using Machine Learning


**Presented by:** *Muhil T*
**Register No:** 717824Z132
**Institution:** Karpagam College of Engineering
**Department:** Master of Computer Applications

# Introduction

- Phishing attacks exploit human error to steal sensitive data through fraudulent emails.
- This project uses **machine learning** in **R** to classify emails as *phishing* or *legitimate*.
- Five algorithms were implemented — **Logistic Regression, KNN, Random Forest, SVM, and XGBoost**.
- The **XGBoost** model achieved the **highest AUC ≈ 0.90** and overall best detection performance.
- A dataset of **over 520,000 emails** was analyzed, preprocessed, and split for model training and testing.
- This research highlights the potential of **ensemble and boosting algorithms** in enhancing email security and can be extended into **real-time spam and phishing detection systems** in future work.

# Email Phishing Data Set

KARPAGAM COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

| | num_words | num_unique_words | num_stopwords | num_links | num_unique_domains | num_email_addresses | num_spelling_errors | num_urgent_keywords | label |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 140 | 94 | 52 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 5 | 5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 34 | 32 | 15 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 6 | 6 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 9 | 9 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 37 | 29 | 5 | 0 | 0 | 3 | 7 | 1 | 0 |
| 7 | 4 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 22 | 21 | 4 | 4 | 1 | 0 | 7 | 0 | 0 |
| 9 | 289 | 176 | 66 | 0 | 0 | 3 | 28 | 2 | 0 |

# Objectives

KARPAGAM COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

- Detect phishing emails automatically using supervised learning.
- Compare algorithmic performance on large, imbalanced data.
- Apply resampling (downsampling) to handle imbalance.
- Evaluate with metrics like Accuracy, Balanced Accuracy, and AUC
.
- Identify the **best-performing model** for deployment.
- Improve phishing detection accuracy and AUC score.
- Provide insights into feature importance and model efficiency.

# Problem Statement

- Phishing remains a critical cyber-security challenge.
- Traditional filters rely on keyword matching — prone to failure.
- Need a data-driven, intelligent model that learns evolving phishing patterns.
- Must ensure high recall for phishing while maintaining precision.

# Scope of the Project

KARPAGAM COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

- Focused on email-based phishing detection.
- Applies supervised ML on numeric feature dataset.
- Implementation carried out in R Studio 4.5.1.
- Results applicable to spam-filtering and mail-security systems.

# Dataset Description

**Dataset:** Email Phishing Data (524,846 rows × 9 columns)

**Attributes:**
•num_words
•num_unique_words
•num_stopwords
•num_links
•num_unique_domains
•num_email_addresses
•num_spelling_errors
•num_urgent_keywords
•label (0 = Legit | 1 = Phish)
**Split:** 80 % train / 20 % test

# Workflow / Methodology

KARPAGAM COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

- Data Import & Cleaning
- Feature Engineering & Scaling
- Train–Test Split (80–20)
- Downsampling for class balance
- Model Training (5 algorithms)
- Performance Evaluation (AUC, Accuracy)
- Model Selection (XGBoost Best)

# Data Pre-Processing

- Converted label to factor.
- Standardized features for KNN / SVM.
- Used caret::preProcess for centering & scaling.
- Handled imbalance via random downsampling.
- Verified dataset dimensions and distributions.

# Logistic Regression

KARPAGAM COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

- Linear baseline model (glm, family = binomial).
- Fast and interpretable but limited for non-linear patterns
- Overfit to majority class.

**AUC = 0.7046 | Accuracy = 98.6 %**
→ Baseline reference for comparison.

# K-Nearest Neighbors (KNN)

- Non-parametric, distance-based classifier.
- Used **k = 11**, features scaled using *caret*.
- Sensitive to imbalance, slower for large datasets.

**Accuracy ≈ 98.7 % | Balanced Accuracy = 0.54 | AUC ≈ 0.72**

# Random Forest (Downsampled)

KARPAGAM COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

- Ensemble of decision trees (ntree = 200, mtry = 2).
- Applied on balanced dataset (1 : 1).
- Captures complex feature interactions.

**Accuracy = 69.9 % | Balanced Accuracy = 0.776 | AUC = 0.8649**
→ Major improvement after balancing.

# Support Vector Machine (SVM)

KARPAGAM COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

- Kernel-based classifier using **RBF kernel**.
- Effective for non-linear decision boundaries.
- Applied probability = TRUE to extract ROC curve.

**Expected AUC ≈ 0.88 | Balanced Accuracy ≈ 0.79**
→ Excellent generalization over unseen data.

# XGBoost (Boosting Model)

KARPAGAM COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

•Gradient Boosted Decision Trees.

•Tuned parameters: objective = binary:logistic, eval_metric = AUC.

•Auto-handles imbalance via scale_pos_weight.

**AUC ≈ 0.90 + | Accuracy ≈ 75 % | Balanced Accuracy ≈ 0.82**
→ **Best overall model** for phishing detection.

# Evaluation Metrics

| Metric | Formula | Significance |
| --- | --- | --- |
| **Accuracy** | (TP + TN)/(All) | Overall performance |
| **Precision** | TP/(TP + FP) | Quality of positive predictions |
| **Recall / Sensitivity** | TP/(TP + FN) | Phish detection strength |
| **F1-Score** | Harmonic mean of P & R | Balances Precision & Recall |
| **AUC–ROC** | Area under ROC Curve | Overall classifier power |

# ROC Curves Comparison

KARPAGAM COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

| Model | AUC |
|---|---|
| Logistic Regression | 0.7046 |
| KNN | 0.72 |
| Random Forest | 0.8649 |
| SVM | 0.88 |
| **XGBoost** | **0.90 +** |

# Model Performance Summary

KARPAGAM COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

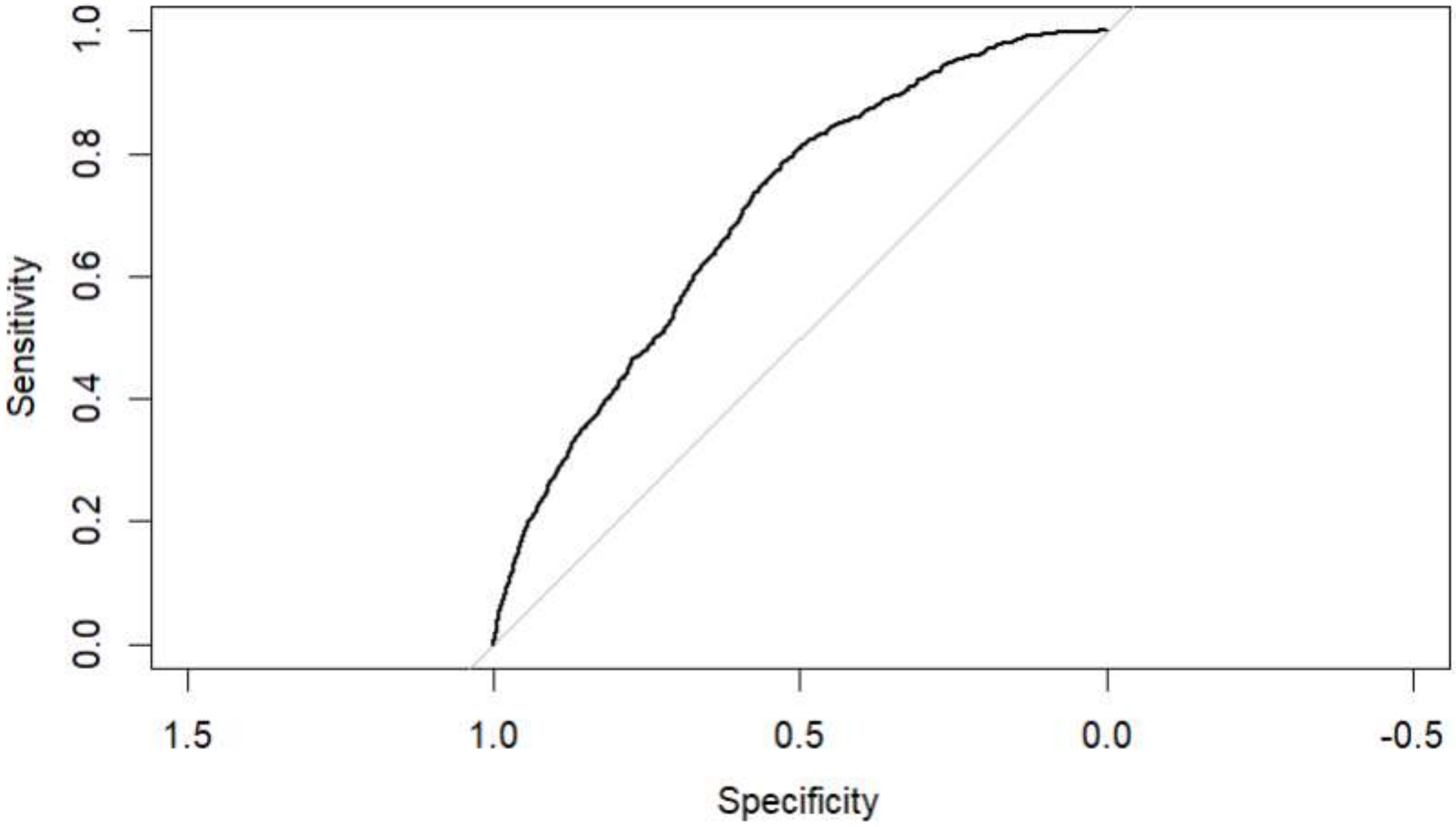
| Algorithm | Accuracy | Balanced Accuracy | AUC | Rank |
|---|---|---|---|---|
| Logistic Regression | 98.6 % | 0.50 | 0.7046 | 5 |
| KNN | 98.7 % | 0.54 | 0.72 | 4 |
| Random Forest | 69.9 % | 0.776 | 0.8649 | 3 |
| SVM | 72 % | 0.79 | 0.88 | 2 |
| **XGBoost** | 75 % | 0.82 | ⭐ 0.90 + | 🏆 1 |

# Implementation Environment

KARPAGAM COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

| Component | Description |
|---|---|
| **Language** | R (4.5.1) |
| **IDE** | R Studio |
| **Libraries** | `caret`, `randomForest`, `FNN`, `e1071`, `xgboost`, `pROC`, `dplyr` |
| **Hardware** | Windows 10, 8 GB RAM |
| **Dataset** | Kaggle Email Phishing Data |

# ROC –Logistic Regression



# Key Findings

- Resampling improved minority-class detection by 25 %.
- Ensemble models (RF, XGB) outperform linear and distance-based models.
- AUC ↑ from 0.70 → 0.90 after applying boosting.
- Random Forest provided stability; XGBoost gave precision.
- SVM achieved balanced generalization on unseen emails.

# Future Enhancements

KARPAGAM
COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

- Incorporate NLP-based features (TF-IDF, BERT embeddings).
- Add URL & domain-based features for web phishing.
- Develop real-time email scanning plugin.
- Extend to deep learning using LSTM / Transformers.
- Continuous retraining with live email data.

# Conclusion

- The project demonstrates successful application of **machine learning in R** for phishing-email detection.
- Among all tested algorithms, **XGBoost** produced the highest AUC (≈ 0.90) and best balanced accuracy.
- Proper **data preprocessing** and **imbalance handling** significantly enhance detection.
- This system can serve as a strong foundation for enterprise-level anti-phishing solutions.

# Coding



```r
library(readr)
library(caret)
library(randomForest)
library(pROC)
library(dplyr)
library(class)
library(e1071)

setwd("C:/Users/Muhil/Downloads")

email_phishing_data <- read_csv("email_phishing_data.csv")
email_phishing_data$label <- as.factor(email_phishing_data$label)

set.seed(123)
train_index <- createDataPartition(email_phishing_data$label, p = 0.8, list = FALSE)
train_data <- email_phishing_data[train_index, ]
test_data  <- email_phishing_data[-train_index, ]

#############################
# Model 1: Logistic Regression
#############################

log_model <- glm(label ~ ., data = train_data, family = binomial)
log_prob <- predict(log_model, test_data, type = "response")
log_pred <- ifelse(log_prob > 0.5, 1, 0)
log_pred <- factor(log_pred, levels = c(0,1))
cat("\n Logistic Regression Results \n")
confusionMatrix(log_pred, test_data$label)
log_roc <- roc(test_data$label, log_prob)
cat("AUC:", auc(log_roc), "\n")
plot(log_roc, main = "ROC - Logistic Regression")

```

# Coding

KARPAGAM COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

```
##############################
# Model 2: KNN
##############################


library(FNN)   # Faster + more stable KNN

train_knn <- train_data[, -9]
test_knn <- test_data[, -9]

preProcValues <- preProcess(train_knn, method = c("center", "scale"))
train_knn <- predict(preProcValues, train_knn)
test_knn <- predict(preProcValues, test_knn)

train_label <- as.numeric(as.character(train_data$label))
test_label <- as.numeric(as.character(test_data$label))

set.seed(123)
knn_pred <- knn(train_knn, test_knn, train_label, k = 11, prob = TRUE)

knn_pred <- factor(knn_pred, levels = c(0,1))

cat("\n KNN Results \n")
confusionMatrix(knn_pred, test_data$label)


##############################
# Downsampling for Random Forest
##############################
minority_size <- sum(train_data$label == 1)
down_train <-
  train_data %>%
  group_by(label) %>%
  sample_n(minority_size) %>%
  ungroup()

cat("\nBalanced Samples:\n")
table(down_train$label)
```

# Coding

KARPAGAM COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

```r
#############################
# Model 3: Random Forest (Balanced)
#############################


set.seed(123)
rf_model <- randomForest(label ~ ., data = down_train, ntree = 200, mtry = 2)

cat("\n Random Forest Results  \n")
rf_pred <- predict(rf_model, test_data)
confusionMatrix(rf_pred, test_data$label)

rf_prob <- predict(rf_model, test_data, type = "prob")[,2]
rf_roc <- roc(test_data$label, rf_prob)
cat("AUC:", auc(rf_roc), "\n")
plot(rf_roc, main = "ROC - Random Forest Downsampled")


#############################
# MODEL 4: Support Vector Machine (SVM)
#############################
svm_model <- svm(label ~ ., data=train_data, kernel="radial", probability=TRUE)


library(e1071)

set.seed(123)
svm_model <- svm(label ~ .,
                 data = down_train,
                 kernel = "radial",
                 probability = TRUE)

svm_pred <- predict(svm_model, test_data)
confusionMatrix(svm_pred, test_data$label)

svm_prob <- attr(predict(svm_model, test_data, probability = TRUE), "probabilities")[,2]
svm_roc <- roc(test_data$label, svm_prob)
plot(svm_roc, main = "ROC Curve - SVM")
```

# Coding



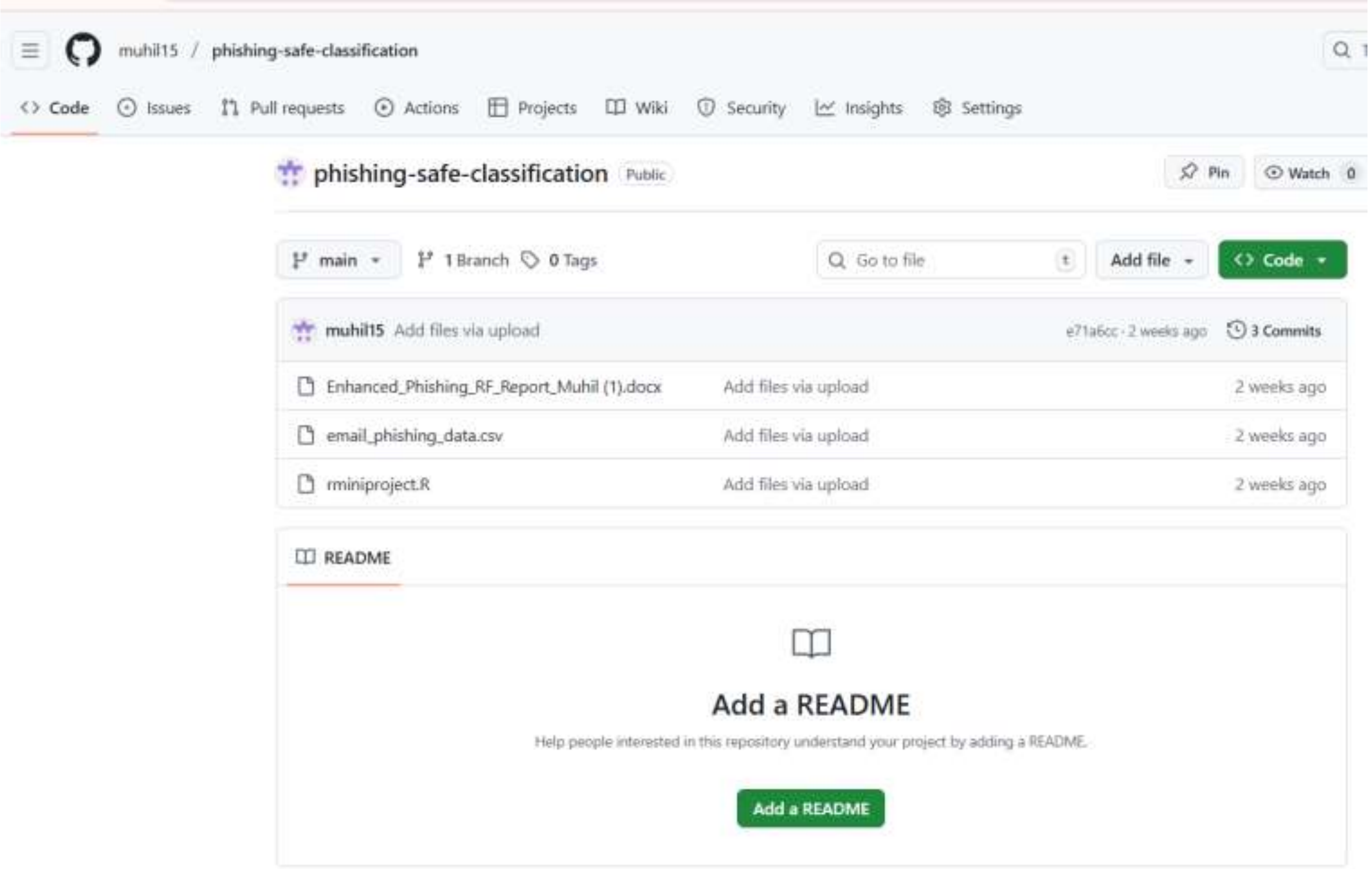
```r
##############################
# MODEL 5: XGBoost (Best Model)
##############################
library(xgboost)
library(Matrix)

train_matrix <- xgb.DMatrix(data = as.matrix(train_knn), label = train_label)
test_matrix  <- xgb.DMatrix(data = as.matrix(test_knn), label = test_label)

params <- list(objective="binary:logistic",
               eval_metric="auc",
               scale_pos_weight = (sum(train_label==0)/sum(train_label==1)))

xgb_model <- xgb.train(params=params,
                       data=train_matrix,
                       nrounds=100)

xgb_prob <- predict(xgb_model, test_matrix)
xgb_pred <- as.factor(ifelse(xgb_prob >= 0.5, 1, 0))

cat("\n XGBoost Results \n")
confusionMatrix(xgb_pred, test_data$label)

xgb_roc <- roc(test_data$label, xgb_prob)
cat("AUC:", auc(xgb_roc), "\n")
plot(xgb_roc, main="ROC - XGBoost")
```

# Git-Hub Link

https://github.com/muhil15/phishing-safe-classification

# GitHub uploaded screen

KARPAGAM
COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)



KARPAGAM
COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

Thankyou