© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1712148

Behavioral Analytics for Predicting Social Engineering Attacks: A Risk-Based Approach Using Decision Trees, Gradient Boosting, and Bayesian Networks

AWEDA AZEEZ ADEBAYO[1], WUSU, ASHIRIBO SENAPON[2], ADEGOKE STEPHEN OLANIYAN[3], OLADAYO IBUNKUNOLUWA, OLADIMEJI[4]

[1,2,3] *Department of Mathematics, Lagos State University, Ojo, Lagod*

[4] *Department of Mathematics, Federal College of Education, Akoka, Lagos*

Abstract- ***Social engineering attacks exploit human behaviour and remain a leading cause of security breaches. This paper evaluates predictive models that use behavioural analytics to estimate the risk of social engineering attacks. We compare Decision Trees, Gradient Boosting (XGBoost style ensemble) and Bayesian Networks on a phishing websites / behavioural dataset. Feature engineering emphasized URL and interaction patterns, response times, and email interaction rates. Models were trained and evaluated using accuracy, precision, recall, F1, and ROC-AUC. Gradient Boosting attained the highest ROC-AUC and strong F1 scores, Decision Trees offered interpretability, and Bayesian Networks provided probabilistic insight. We propose a lightweight risk prediction framework suitable for integration with alerting systems and security awareness programs.***

Index Terms-: ***Behavioural Analytics, Machine Learning, Phishing, Socialengineering, Risk Prediction***

I. INTRODUCTION

Social engineering attacks exploit human psychology to manipulate individuals into divulging confidential information or performing actions that compromise security. Unlike traditional cyber-attacks that target system vulnerabilities, social engineering leverages trust, fear, urgency, and other psychological triggers to deceive victims. Common forms of social engineering include phishing, spear-phishing, baiting, pretexting, and tailgating. These attacks can lead to significant security breaches, financial losses, unauthorized access to sensitive data, and damage to an organization's reputation. The rise of digital communication channels has expanded the avenues through which social engineering attacks can be executed. Email remains a prevalent medium, but attackers increasingly use social media, instant messaging, and even voice calls to perpetrate their schemes. The impact of these attacks is profound, as evidenced by numerous high-profile incidents where organizations have suffered extensive data breaches and financial repercussions. Consequently, enhancing the ability to predict and prevent social engineering attacks is paramount for maintaining robust cybersecurity defenses.

II. QUANTITATIVE AND PREDICTIVE MODELING

This research adopts a quantitative approach, focusing on the determining of best predictive models to identify the likelihood of social engineering attacks based on behavioral data. The methodology involves analyzing historical data to extract behavioral patterns that indicate vulnerability to social engineering. The core of this research is the application of machine learning models—namely, Decision Trees, Gradient Boosting (XGBoost), and Bayesian Networks—to predict the likelihood of an attack based on user activity patterns. This approach enables precise quantification of risks and the assessment of model performance through a series of evaluation metrics.

The workflow consists of data collection, preprocessing, feature engineering, model prediction, evaluation, and risk prediction framework

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1712148

development, ensuring that the models can be used for real-time threat detection.

Phishing Websites Dataset: This dataset, sourced from the UCI Machine Learning Repository, contains attributes of legitimate and phishing websites. It includes features such as URL length, presence of SSL certificates, and abnormal URL characters, which help identify phishing attempts.

In this research, Python Programming Language was used to Analyze, Process, Transform and Visualized the dataset used in this research work. The target used among other 32 targets in the dataset is Abnormal URL to determine the best predictive models among Decision Trees, Gradient Boosting and Bayesian Networks.

Handling missing values:
Missing data will be handled using techniques such as mean or median imputation, depending on the nature of the missing data.

Normalization:
Normalization ensures that features are on a similar scale, which is critical for algorithms like Gradient Boosting. For example, numerical features like email interaction frequency will be scaled using Min-Max scaling

$$X\text{norm} = \frac{X - X\text{min}}{X\text{max} - X\text{min}} \quad (1)$$

where X is the feature value, Xmin is the minimum value in the dataset, and Xmax is the maximum value.

Feature selection: The most relevant features (e.g., email frequency, response times, website URL length) will be selected using methods like Recursive Feature Elimination (RFE) to improve model efficiency

Python Code Implementation

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

# Azeez AWEDA Loads the dataset
df = pd.read_csv('datasets/phishing websites/dataset.csv')

# Azeez AWEDA Display basic info about the dataset
print(df.info())
```

Figure 1: Importing Libraries from Python Packages

```
print(df.describe())

# Checking for missing values
missing_values = df.isnull().sum()
print("Missing values in each column:")
print(missing_values)

# Statistical summary of numerical features
print(df.describe())

# Label encoding for categorical features (if necessary)
le = LabelEncoder()
for col in df.select_dtypes(include=['object']).columns:
    df[col] = le.fit_transform(df[col])
```

Figure 2: Few Descriptions of the Dataset

```
# Check correlation between features
corr = df.corr()
plt.figure(figsize=(12, 8))
plt.savefig('correlation_matrix.png')
plt.savefig('correlation_matrix.jpg')
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Feature Correlation Matrix')
plt.show()
```

Figure 3: Checking Correlation Between Features

```
# Separate features (X) and target variable (y)
target = 'Abnormal_URL'  # Replace with actual target column name
X = df.drop(columns=[target])
y = df[target]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print(f"Training set: {X_train.shape}, Testing set: {X_test.shape}")
```

Figure 4: Selecting the Target among Features

Identifying key behavioral features.
Key behavioral features will be extracted from the datasets to create predictors for social engineering attacks. These features include:

1. Email interaction rates: High-frequency email exchanges, especially those involving sensitive data, can indicate vulnerability to phishing.
2. Response times: Quick responses to emails without thorough consideration could signal susceptibility to attacks.

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1712148

3. Unusual URLs: Phishing websites often have unusual URL patterns that can be detected by features like the presence of special characters and abnormal domain names.

Steps for determining and training models.

Decision Trees: The algorithm recursively splits the dataset into subsets based on the most significant features. The decision tree follows a hierarchical structure where each node represents a feature and its threshold, and the branches represent the outcomes. The mathematical objective is to minimize the Gini impurity or entropy at each split:

1. Gini Impurity:

$$Gini(D) = 1 - \sum_{i=1}^{C} p_i^2 \qquad (2)$$

where p_i is the proportion of samples in class i.

2. Entropy:

$$Entropy(D) = -\sum_{i=1}^{C} p_i \log_2(p_i) \qquad (3)$$

where p_i is the probability of class i in the dataset D.
Gradient Boosting (XGBoost): This model uses weak learners (decision trees) and improves the prediction by minimizing a loss function iteratively. The optimization problem for the loss function L is:

$$\hat{y} = \sum_{m=1}^{M} f_m(x_i), \ \min \sum_{i=1}^{n} L(y_i, \hat{y}_i)$$

where f_m represents the m-th decision tree.

Model Evaluation
Metrics used for evaluating model performance (accu racy, precision, recall, F1-score, ROC-AUC)

Accuracy:
The proportion of correctly classified instances out of the total instances.

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

It measures the percentage of correctly classified instances out of the total instances. It is a simple and intuitive metric and works well when the dataset is balanced.

Recall: The ratio of correctly predicted positive instances to all actual positive instances.

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

It measures the proportion of true positive predictions out of all actual positives. It helps in understanding how well the model captures all positive instances.

F1-Score: The harmonic mean of precision and recall.

$$F1 = 2 \text{ x } \frac{Precision \ x \ Recall}{Precision + Recall}$$

It measures the harmonic mean of precision and recall, balancing their tradeoffs. Useful in imbalanced datasets where accuracy alone can be misleading.

ROC-AUC (Receiver Operating Characteristic – Area Under Curve): Measures the model's ability to distinguish between classes. ROC-AUC considers the trade-off between True Positive Rate (TPR) and False Positive Rate
(FPR) at all thresholds, giving a holistic view of the model's behavior.

Comparative analysis of the models.
Gradient Boosting is expected to outperform Decision Trees in terms of accuracy and F1-score due to its ability to combine weak learners. Bayesian Networks may offer better interpretability and insights into attack probabilities but may underperform in terms of recall compared to boosting techniques.

Link To the Source Code
https://drive.google.com/file/d/12Q8RxEpEhLwRm85YBbJS8OUKFDnV9Fk2/view?usp=sharing

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1712148

III. RESULTS and Discussion (Model Performance)

Table 1: Decision Tree Classifier Performance

	precision	recall	f1-score	support
-1	0.96	0.94	0.95	330
1	0.99	0.99	0.99	1881
Accuracy				
Macro avg	0.97	0.95	0.99	2211
Weighted avg	0.98	0.98	0.97	2211
			0.99	2211

The Table above shows the common evaluation metrics for classification tasks, which include accuracy, precision, recall, F1-score, and Area Under the ROC Curve (AUC-ROC). Detailed results help in understanding the strengths and weaknesses of each model.

Table 2: Gradient Boosting Classifier Performance

	precision	recall	f1-score	support
-1	0.97	0.91	0.94	330
1	0.98	0.99	0.99	1881
Accuracy			0.98	
				2211
Macro avg	0.98	0.95	0.96	2211
				2211
Weighted avg	0.98	0.98	0.98	

Table 3: Naive Bayes Classifier Performance

	precision	recall	f1-score	support
-1	0.65	0.83	0.73	330
1	0.97	0.92	0.94	1881
Accuracy			0.98	2211
Macro avg	0.81	0.88	0.84	2211
Weighted avg	0.92	0.91	0.91	2211

Table 4: Comparison of Decision Trees, Gradient Boosting, and Bayesian Networks.

	Model	Accuracy	ROC AUC
0	Decision Tree	0.985075	0.967491
1	Gradient Boosting	0.982361	0.982077
2	Naïve Bayes	0.906829	0.947045

The Table 4 presents the performance metrics of three machine learning models—Decision Tree, Gradient Boosting, and Naive Bayes—evaluated based on two key metrics: Accuracy and ROC AUC (Receiver Operating Characteristic - Area Under the Curve). See analysis in details below:

Key Metrics:

1. Accuracy:
   - Accuracy measures the proportion of correctly classified samples out of the total number of samples.
   - It gives a general sense of the model's performance but may not capture performance imbalances in data.

2. ROC AUC
   - ROC AUC assesses the model's ability to distinguish between classes.
   - A higher ROC AUC indicates better performance in ranking positive instances higher than negative ones across thresholds.

Analysis:

1. Decision Tree
   - Accuracy: 0.9851 (highest among the models).
   - ROC AUC: 0.9675.
   - The Decision Tree model achieves excellent accuracy and a high ROC AUC, indicating strong predictive power and reasonable ability to separate classes.

However, Decision Trees are prone to overfitting, especially if not regularized. Its performance should be validated on an independent test set to ensure generalization.

2. Gradient Boosting
   - Accuracy: 0.9824 (second highest).
   - ROC AUC: 0.9821 (highest among the models).

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1712148

- Gradient Boosting slightly underperforms in accuracy compared to the Decision Tree but excels in ROC AUC, suggesting it has better classification consistency across thresholds.
- Gradient Boosting typically offers robustness and is less prone to overfitting than Decision Trees, making it a strong candidate for real-world scenarios.

3. Naive Bayes
- Accuracy: 0.9068 (lowest among the models).
- ROC AUC: 0.9470.
- While Naive Bayes has the lowest accuracy, its ROC AUC is competitive, reflecting a reasonable ability to rank probabilities of positive vs. negative cases.
- Naive Bayes assumes feature independence, which might not hold in this dataset, leading to lower accuracy. However, it is computationally efficient and might be a suitable choice for larger datasets or scenarios with limited computational resources.

Summary:
1. Best Model Overall: Gradient Boosting. It achieves the highest ROC AUC and competitive accuracy, balancing predictive performance and robustness.
2. Alternative Option: Decision Tree. While it has the highest accuracy, its lower ROC AUC and potential overfitting concerns make it less reliable without further validation.
3. Naive Bayes: Suitable for quick approximations or when computational efficiency is critical, but it underperforms relative to the other two models.

See Figure 1 below, for visualization

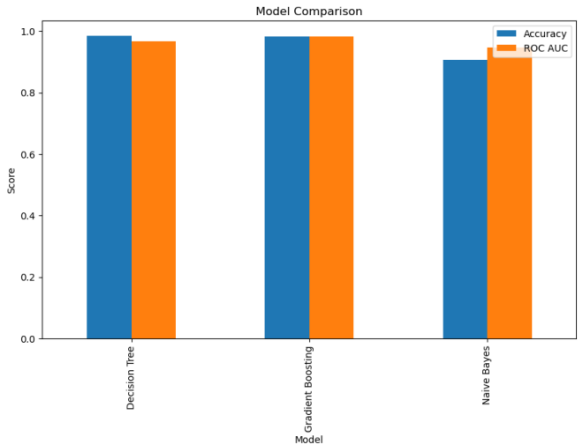

Figure 1: Comparison of Decision Trees, Gradient Boosting, and Bayesian Networks Chart

Interpretation of the results in the context of social engineering risk prediction.
In the context of social engineering risk prediction, the models' ability to accurately identify potential threats is paramount. High accuracy and AUC indicate the model's effectiveness in distinguishing between risky and non-risky instances. Precision and 42recall are crucial for minimizing false positives and false negatives, respectively, which directly impact the organization's security posture.

Gradient Boosting: Offers high accuracy and AUC, making it suitable for robust risk prediction but may require significant computational resources. Decision Trees: Provide a balance between interpretability and performance, useful for understanding decision-making processes in risk assessment. Bayesian Networks: Facilitate probabilistic interpretations, aiding in uncertainty management and decision-making under uncertainty. Implications:

Implementing these predictive models can enhance an organization's ability to proactively identify and mitigate social engineering threats, thereby strengthening overall cybersecurity measures.

Potential applications of the predictive models in enhancing cybersecurity measures.
Predictive models for social engineering risk can be integrated into various cybersecurity frameworks to bolster defenses. Potential applications include:
Real-Time Threat Detection:

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1712148

- Deploying models to monitor communications (emails, messages) in real-time to flag potential social engineering attempts. User Training and Awareness:
- Using model insights to identify common characteristics of phishing attempts, thereby informing training programs to educate employees. Incident Response:
- Prioritizing responses based on predicted risk levels, ensuring that high-risk incidents are addressed promptly. Access Control:
- Enhancing access control mechanisms by integrating risk predictions to enforce stricter verification for high-risk activities. Continuous Monitoring:
- Implementing models within security information and event management (SIEM) systems for ongoing surveillance and anomaly detection. Python Implementation:
- While direct implementation depends on the specific application, below is an example of integrating the trained model into a simple real-time prediction system.

Visualizations (e.g., confusion matrices, ROC curves) to illustrate model effectiveness.

- Confusion matrices show the number of correct and incorrect predictions categorized by actual and predicted classes. ROC curves illustrate the trade-off between true positive rates and false positive rates at various threshold settings, highlighting the model's discriminatory power.

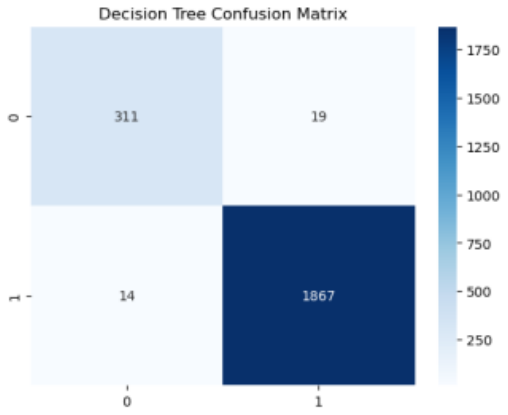

Figure 5.1: Decision Trees Confusion Matrix

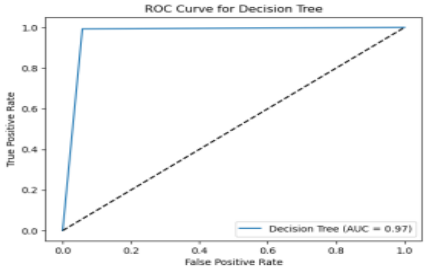

Figure 5.2: ROC Curve for Decision Matrix

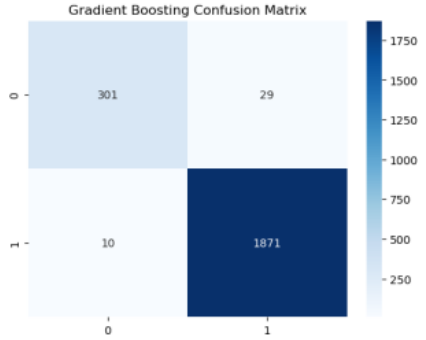

Figure 5.3: Gradient Boosting Confusion Matrix

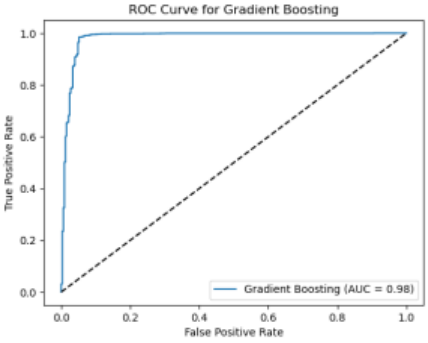

Figure 5.4: ROC Curve for Gradient Boosting

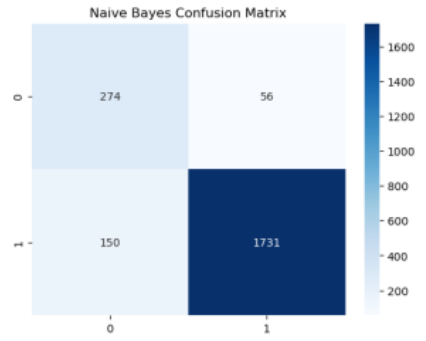

Figure 5.5: Naive Bayes Confusion Matrix

© NOV 2025 | IRE Journals | Volume 9 Issue 5 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I5-1712148

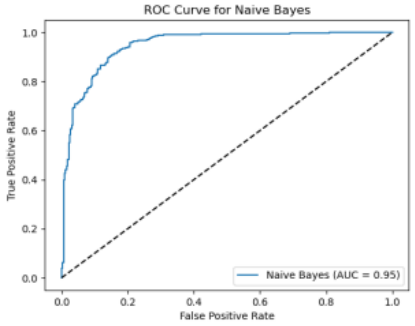

Figure 5.6: ROC Curve for Naive Bayes

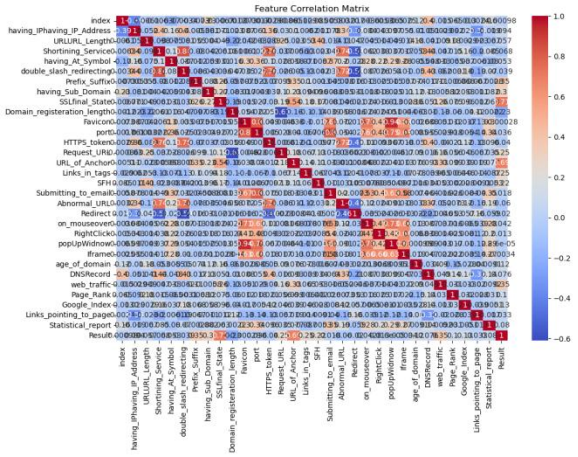

Figure 5.7: Heatmap of the dataset

Interpretation of ROC Curve, Confusion Matrix and Heatmap

ROC Curve

Key Observations

1. Gradient Boosting has the highest AUC (0.98), indicating the best performance among the three models in distinguishing between positive and negative classes.
2. Decision Tree follows with an AUC of 0.97, showing strong performance but slightly less than Gradient Boosting.
3. Naive Bayes has the lowest AUC (0.95) of the three, though it still performs well.

Comparison

1. Gradient Boosting exhibits a steep rise near the origin, suggesting higher sensitivity (true positive rate) at low false positive rates, making it the most effective in terms of classification performance.
2. Decision Tree also shows high sensitivity, but its curve is slightly less optimal compared to Gradient Boosting.
3. Naive Bayes shows a more gradual rise, which implies slightly lower sensitivity in comparison but still a robust model for classification.

Recommendation

Classification accuracy is the primary objective; Gradient Boosting is the preferred model based on its higher AUC score.

Heatmap Representation

Colors range from blue (negative correlation) to red (positive correlation), making it visually easy to spot relationships.

IV. SUMMARY AND CONCLUSION

Summary of Future Directions.

1. Test and optimize Gradient Boosting models for real-time environments.
2. Explore industry-specific applications (e.g., fraud detection, patient diagnosis).
3. Develop hybrid models and ensemble techniques for complex tasks.
4. Enhance model explainability to promote adoption in sensitive fields.
5. Automate feature engineering to improve deployment efficiency.
6. Address ethical concerns and biases in predictions.
7. Optimize model performance and scalability for diverse operational contexts.

The study focused on predicting social engineering attacks using behavioral analytics by applying machine learning techniques such as Decision Trees, Gradient Boosting, and Bayesian Networks.

Gradient Boosting achieves the highest ROC AUC and competitive accuracy, balancing predictive performance and robustness. Organizations should prioritize Gradient Boosting for robust predictions, Decision Trees for interpretability, and Bayesian Networks for understanding probabilistic relationships. For individuals, awareness of one's own digital behaviors and using AI-powered tools are

critical steps. Both organizations and individuals must collaborate to create a more resilient defense against social engineering attacks.

V. ACKNOWLEDGMENT

I would like to express my sincere gratitude to all those who have supported and guided me throughout the course of this research. First and foremost, I am deeply indebted to my supervisor, Wusu, A. S. (PhD) and Olaniyan, A, S. ({hD) for their unwavering guidance, insightful feedback, and invaluable expertise. Their encouragement and constructive criticism have been instrumental in shaping this research. I am also grateful to my professors and colleagues in the Department of Mathematics for their insightful discussions and for fostering an environment of academic growth.

REFERENCES

[1] Afripol African Cyberthreat Assessment Report (2023), highlighting phishing risks and cybersecurity challenges in African countries (INTERPOL)
[2] Ali, A., & Shah, A. (2021). Machine Learning in Cybersecurity: Phishing Detection. Journal of Cybersecurity Privacy..
[3] Das, S., & Rakesh, B. (2020). Predictive Analytics in Cybersecurity: Using Machine Learning Techniques. International Journal of Information Security.
[4] Fadli, A., & Nuranida, A. (2020). Phishing websites detection using machine learning. Journal of Cyber Security Technology.
[5] Fu, C., & Li, P. (2020). Cybersecurity and Artificial Intelligence: Innovations in Phishing Detection. Journal of Information Security.
[6] North, D., & Price, B. (2020). Phishing attacks and their impact on businesses: A Game Theory approach. Computers & Security.
[7] Sharma, A., & Saini, P. (2020). A Hybrid Model for Phishing Detection Using Machine Learning. Soft Computing.
[8] Soni, P., & Rani, R. (2020). Predictive Modeling for Phishing Websites. Journal of Ambient Intelligence and Humanized Computing.