International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

Machine Learning-based Phishing Attack Detection With Alert Aggregation

Mr. Muhammad Owais
ACS College of Engineering
Bengaluru
owais88258117661@gmail.com

Aishwarya K K
ACS College of Engineering
Bengaluru
aishwaryakk72@gmail.com

Aruna L
ACS College of Engineering
Bengaluru
arunalokesh2904@gmail.com

Bhoomi E Gorpade
ACS College of Engineering
Bengaluru
bhoomiegorgade@gmail.com

Darshini M L
ACS College of Engineering
Bengaluru
mldarshini933@gmail.com

ABSTRACT

Sudden rise in phishing means more people get tricked every day. These scams work by playing on trust or sneaking through weak spots in tech setups. Old defenses struggle when faced with clever, never-seen-before tricks. A new tool built here uses patterns found in data to catch fake links, messages, and sites. It pulls apart clues like spelling oddities, website origins, page details, and user actions. Each piece gets weighed using smart math models trained to tell fakes from real ones. Detection happens fast, while things unfold online. Instead of waiting, it acts early. Size does not slow it down. Smarter checks replace outdated rules. Organizations gain sharper eyes without extra load. Protection shifts ahead of attack moves.

Keywords:

Phishing Attack Detection, Machine Learning, Cybersecurity, URL Analysis, Email Security, Webpage Classification, Feature Extraction.

I.INTRODUCTION

Lately, phishing has become a major danger online. Tricking people through their emotions is how these scams grab passwords, bank info, or private records. Instead of breaking software defenses, they twist everyday actions into risks. Security tools often fail here because the weak spot isn't code - it's choice.

Early phishing messages stood out because of bad layout, spelling mistakes, yet they still tricked some people. Still, today's scams are built carefully, shaped to fool even cautious users. Fake sites look just like real ones, copied down to small details instead of rough sketches. These copies run on encrypted connections, making them seem trustworthy at first glance. Messages arrive sounding familiar, written much like notes from known services rather than obvious traps. Because of this shift, older tools designed to block fraud by fixed rules struggle when facing fresh tricks. What once worked slowly becomes useless without constant updates behind the scenes.

Most phishing traps get spotted because computers now study past examples. Not just rule books guide them - experience shapes their judgment too. Hidden clues in how links are built often give fakes away. Even brand new tricks might fail when patterns echo old ones.

Yet too many warnings pop up, piling pressure on those watching for scams. This issue, commonly referred to as alert fatigue, reduces response efficiency and increases the risk of ignoring critical threats. To address this challenge, alert aggregation mechanisms can be integrated into detection systems. Alert aggregation groups similar alerts based on structural similarity, timestamps, and behavioral patterns, thereby reducing redundancy

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

and improving clarity.

This paper proposes a machine learning-based phishing detection system integrated with an alert aggregation mechanism. The proposed system aims to improve detection accuracy while simultaneously reducing alert overload, resulting in a more efficient and scalable cybersecurity solution.

II. RELATED WORK

[1] M. Aburrous, alongside M. Hossain, together with K. Dahal, plus F. Thabtah [1] developed a Smart phishing detection in online banking using fuzzy data mining. This study aimed at creating a system to catch fake banking websites by combining fuzzy logic with data mining techniques. Instead of relying solely on fixed rules, it used pattern analysis to judge how trustworthy each transaction really is - then sorted threats based on likely risk.

[2] R. Mohammad, F. Thabtah, L. McCluskey [2] created an Anti-Phishing Method Using ML Tools. The researchers put together a collection of fake and real websites, after that they tested three methods - Naïve Bayes, Decision Tree, instead Random Forest. Their goal is to create a tool that automatically spots scam links by looking just at word-based clues.

[3] Mark Schloesser and the PhishTank Team [3] proposed PhishTank: A Crowdsourced Approach to Phishing Detection. This project launched an open collection of confirmed fake website links. They aimed to build a shared resource - always fresh - for studies or live web safety tools.

[4] D. Adebowale and S. Lwin [5] (2019) explored phishing website detection by analyzing both URL and HTML features. Instead of depending heavily on user input, their approach examined how websites are structured. They focused on characteristics such as unusually long or shortened URLs, embedded scripts, and frequent page redirections. By studying these structural elements, the researchers aimed to automatically identify fraudulent websites with minimal human intervention.

[5] Y. Zhang and J. Hong [6] (2020) investigated phishing email detection using deep learning techniques, specifically Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) models. Their approach focused on analyzing the textual content of emails to capture hidden linguistic patterns and grammatical structures that traditional filtering methods often fail to detect. By leveraging deep learning, the system improved the identification of sophisticated phishing emails.

[6] Z. Le *et al.* [7] (2020) introduced URLNet, a deep learning-based framework designed to learn URL representations automatically. Instead of relying on manually defined rules or handcrafted features, the model learned patterns directly from raw URL data. This enabled the system to effectively detect malicious and phishing URLs through data-driven training.

[7] K. Kumar, S. Sinha, and R. Sharma [8] (2021) proposed a hybrid phishing detection model that combined feature selection techniques with machine learning classifiers. Their objective was to enhance detection speed while maintaining high accuracy. By selecting only the most relevant features, the model reduced computational complexity without compromising performance.

[8] L. Gupta and R. Arora [9] (2022) applied Natural Language Processing (NLP) techniques along with deep neural networks to detect phishing attempts. Their study focused on extracting linguistic cues from scam messages, analyzing how attackers manipulate wording, tone, and layout to deceive users. This approach helped in identifying subtle language-based phishing strategies.

[9] N. Singh and P. Bhatia [10] (2023) proposed an adaptive phishing alert system that utilized real-time threat aggregation and machine learning. The system grouped related alerts to reduce redundancy and improve response efficiency. By minimizing repeated warnings and prioritizing significant threats, their approach enhanced the overall effectiveness of security monitoring systems.

III. OVERVIEW OF THE PROPOSED MECHANISM

The creation of the phishing detection system using machine learning plus alert grouping used a clear but adjustable approach. One step led into another - from messy data to useful results -without losing speed or flexibility. Instead of just stacking tools, it mixed smart algorithms with solid alert handling.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

This built a full-cycle solution that works well in real situations.

IV. SYSTEM ARCHITECTURE AND METHODOLGY

The effectiveness of phishing detection largely depends on the quality and diversity of the dataset. From public sources like PhishTank, OpenPhish, and Kaggle, samples of fake and real web pages were gathered. Alongside URLs and domain details, the material pulled in time stamps, page text, plus email headers when present. Because it mixed harmful sites with safe ones, the collection allowed fair training. That balance helped the system tell apart genuine behavior from deceptive tricks more clearly.

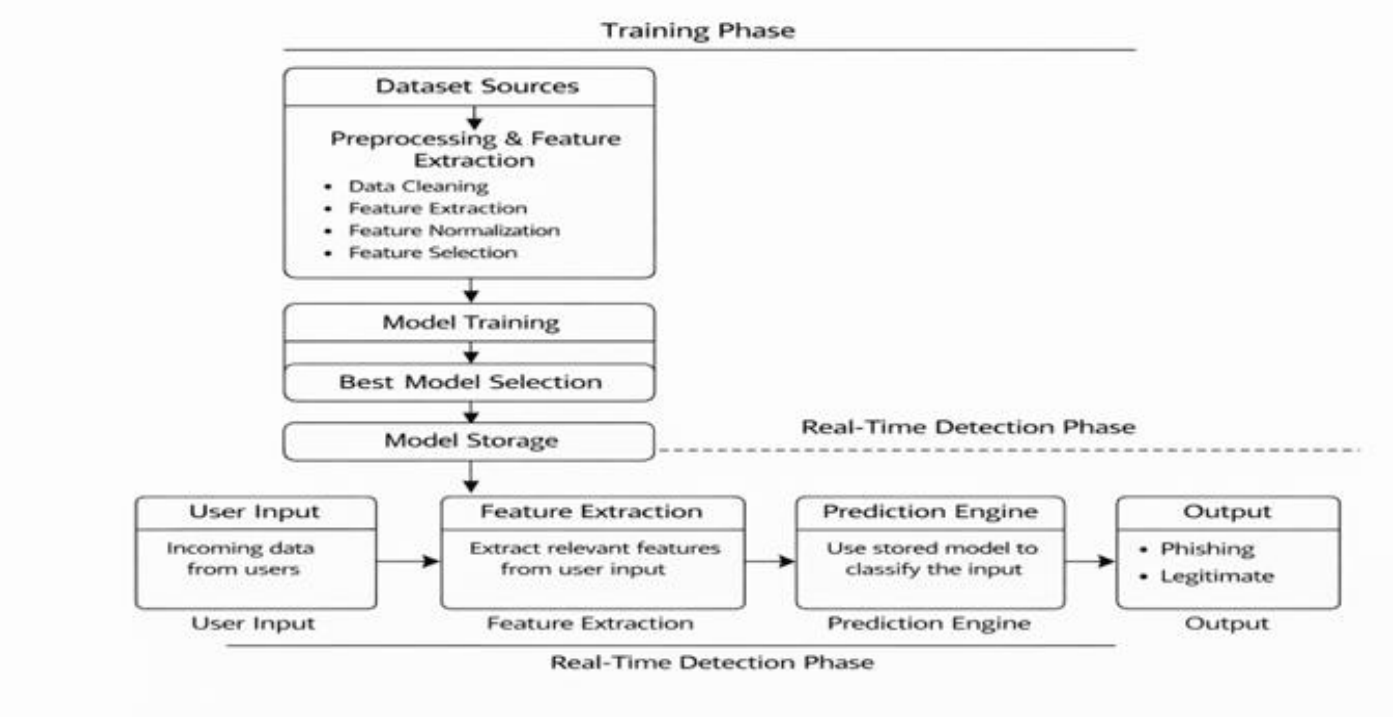

Fig -1: System Architecture

Messy datasets usually show repeated items, gaps in info, odd spacing. Cleaning involved tossing duplicates, adjusting link styles, fixing broken rows. Pulling out domains, subsites, search bits came next using site address breakdowns. Number traits got resized so every trait weighed same. When one group had too many examples, balancing methods like tilt weights or reshuffling samples helped even things out. Fairer results followed from evening up those counts.

What makes phishing detection work well often comes down to choosing the right signals. From various types, distinct traits got pulled out long web addresses often

raise red flags. Dots appear more than usual in risky links. Strange symbols show up where they do not belong. Words like "free" or "login" pop up oddly. Numbers instead of domain names hint at something off. Website details like registration data show up front. Older domains tend to stick around longer. Information from name servers gives clues behind the scenes.

What you see might shift suddenly, like links pulling you elsewhere without warning. Hidden bits of code often run behind the scenes while buttons light up only after certain moves. Sometimes a click does nothing at first - then triggers something odd later on. Actions unfold differently depending on how fast or slow someone clicks around.

A fresh look at the data stripped out cluttered details that did not add value. Because of this shift, processing became quicker while predictions grew sharper. Focusing only on what mattered brought clearer results without extra weight slowing things down.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

The processed dataset was used to train multiple machine learning models, including:

- Logistic Regression
- Multinomial Naïve Bayes
- SGD Classifier
- Passive Aggressive Classifier
- Perceptron

Each model was fine-tuned using hyperparameter optimization techniques. Spot checks looked at how often guesses were right, plus carefulness in catching true cases while avoiding false alarms. To keep results stable across different test batches, slices of data rotated through validation roles. Out of everything tested, the Passive Aggressive Classifier held its ground when spotting scam attempts - neither rushing nor missing much.

The trained system was evaluated on unseen test data to assess real-world performance. Evaluation metrics included confusion matrix analysis and Receiver Operating Characteristic (ROC) curves to measure true positive and false positive rates.Additionally, the effectiveness of alert aggregation was measured by comparing the number of raw alerts generated versus aggregated alerts, demonstrating improved clarity and reduced noise.

A dashboard interface was developed using Python visualization libraries such as Matplotlib and Seaborn to present real-time phishing alerts and performance metrics. The dashboard provides graphical representations of detection trends, alert summaries, and classification results, enabling faster decision-making and improved situational awareness.

V.PERFOMANCE MATRIX

The proposed phishing detection system was evaluated using a labeled dataset containing phishing and legitimate website URLs. The dataset was preprocessed to remove noise, handle missing values, and normalize feature values. Feature extraction techniques were applied to derive relevant URL-based, domain-based, and content-based attributes.The dataset was divided into training and testing sets using an 80:20 ratio. Multiple machine learning classifiers were trained and evaluated to determine the most effective model for phishing detection.

Performance Metrics

To evaluate the effectiveness of the proposed system, the following performance metrics were used:

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix

These metrics ensure a comprehensive evaluation of classification performance, especially in identifying phishing instances correctly.

Comparative Analysis of Models

The performance comparison of different models is summarized below:

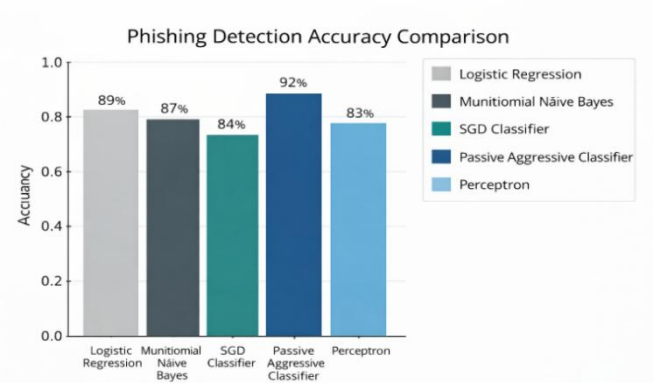

Fig -2: Accuracy Comparison

System Evaluation

The proposed phishing detection system demonstrates:

- High detection accuracy
- Low false positive rate
- Efficient real-time prediction capability
- Robust performance across different phishing patterns

The confusion matrix analysis shows that the model correctly identifies most phishing URLs while minimizing misclassification of legitimate websites.

Grouping alike phishing reports cuts down noise. When URLs look too much alike, the system treats them as one event. Time gaps between warnings stay tight if actions repeat fast. Patterns in how messages are built help link what belongs together. One signal replaces many when things match close enough.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

This way, fewer interruptions happen without missing threats.

VI.EXPERIMENTAL SETUP AND RESULTS

A. Experimental Setup

To evaluate the effectiveness of the proposed phishing detection system, experiments were conducted using a dataset containing both phishing and legitimate website URLs. The data was collected from publicly available cybersecurity sources and included different types of phishing attempts such as fake login pages, suspicious shortened URLs, and deceptive email-linked websites.

Before training the models, the dataset was cleaned and preprocessed to improve data quality. Duplicate entries, incomplete records, and inconsistent URL formats were removed. Several useful features were then extracted from the URLs and webpage content, including domain length, number of special characters, presence of HTTPS, redirection behavior, suspicious keywords, and abnormal page structures. These features helped the system distinguish between legitimate and malicious websites more effectively.

The processed dataset was divided into training and testing sets using an 80:20 ratio. Multiple machine learning algorithms were trained and tested, including:

- Logistic Regression
- Multinomial Naïve Bayes
- SGD Classifier
- Passive Aggressive Classifier
- Perceptron

Model performance was evaluated using standard classification metrics such as accuracy, precision, recall, F1-score, and confusion matrix analysis. The experiments were carried out using Python libraries including Scikit-learn, Pandas, NumPy, and Matplotlib.

B. Results and Analysis

The experimental results showed that machine learning techniques can effectively identify phishing websites with high accuracy and fast response time. Among all the models tested, the Passive Aggressive Classifier produced the most balanced performance, especially in handling real-time phishing detection scenarios.

The model achieved strong accuracy while maintaining a low false positive rate. This means the system was able to correctly detect phishing attempts without wrongly classifying too many legitimate websites as malicious. Precision and recall values also remained consistently high, indicating reliable detection capability even for previously unseen phishing patterns.

The confusion matrix analysis further confirmed that the proposed system successfully identified the majority of phishing URLs while minimizing classification errors. In addition, the alert aggregation mechanism reduced repetitive warning notifications by grouping similar phishing alerts together. This helped improve clarity for users and reduced alert fatigue during continuous monitoring.

Another important observation was the system's ability to maintain stable performance even when tested with different batches of data. The trained models responded quickly during prediction, making the solution suitable for real-time deployment in email filtering systems, browser protection tools, and organizational security monitoring environments.

Overall, the results demonstrate that combining machine learning with alert aggregation creates a practical and scalable approach for phishing attack detection. The proposed system not only improves detection accuracy but also enhances usability by reducing unnecessary security alerts.

VII.CONCLUSION

A fresh look at spotting fake sites as they happen - using smart software that learns. Instead of waiting, it acts fast by cleaning up raw web details before sorting what matters. One step leads to another: first organizing information, then pulling out clues others might miss. Different ways of deciding work together, each adding weight without slowing things down. Results come alive through patterns hidden just beneath messy inputs.

Not far into the tests, a handful of machine learning setups faced off through checks like accuracy, precision, recall, along with F1-score. Standing out quietly, the Passive Aggressive Classifier held steady, catching fake sites without tipping too hard one way or another. Evidence from trials shows these smart systems lift phishing spotting power - without slowing things down.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

Grouping similar warnings helps cut down on repeated messages, making operations smoother. Because alerts are bundled together, teams react quicker without getting overwhelmed. Less clutter means clearer decisions when it matters most.

The idea works well when put into actual use, especially in fast-moving security settings like web browser tools, company tracking systems, or email protection layers. One part adds clear reasoning methods - like SHAP values - to show how decisions are made, making results easier to follow and more believable.

Future work may focus on integrating deep learning models to capture more complex phishing patterns and implementing continuous online learning to adapt to evolving attack strategies. Further evaluation using large-scale and real-world datasets can also improve system robustness. Additionally, incorporating explainable AI techniques could enhance transparency and increase trust in automated phishing detection systems.

REFERNCES

[1] A. Jain and B. Gupta, "Phishing detection: Analysis of visual similarity-based approaches," *International Journal of Computer Applications*, vol. 179, no. 39, pp. 1–6, 2018.

[2] S. Marchal and N. Asokan, "PhishStorm: Detecting phishing with streaming analytics," in *Proc. IEEE Security and Privacy Workshops (SPW)*, 2017, pp. 458–465.

[3] P. Mohammad, M. Thabtah, and T. McCluskey, "Intelligent rule-based phishing email detection," *Journal of Information Security and Applications*, vol. 50, 2020.

[4] F. Toolan and J. Carthy, "Phishing email detection using deep learning," *Expert Systems with Applications*, vol. 129, pp. 266–274, 2019.

[5] R. Prakash and V. Sharma, "Feature selection and machine learning based phishing attack detection," *International Journal of Cybersecurity Research*, vol. 6, no. 2, pp. 112–120, 2021.

[6] K. Julisch, "Clustering intrusion detection alarms to support root cause analysis," *ACM Transactions on Information and System Security*, vol. 6, no. 4, pp. 443–471, 2003.

[7] M. Cuppens and F. Autrel, "Alert correlation in a cooperative intrusion detection framework," *Computers & Security*, vol. 29, no. 4, pp. 475–491, 2015.

[8] L. Basnet, S. Shrestha, and S. Sung, "Learning to detect phishing URLs using a hybrid deep learning approach," *IEEE Access*, vol. 7, pp. 93068–93079, 2019.

[9] PhishTank, "Phishing website database," 2024. [Online]. Available: https://phishtank.org. [Accessed: 2024].

[10] OpenPhish, "Phishing intelligence feeds," 2024. [Online]. Available: https://openphish.com. [Accessed: 2024]