Adoption of AI-powered Phishing Detection through Lightweight Machine Learning Models to Secure Web Traffic in the Real World

Ronakbhai B. Bhalala [1]

Senior Project Associate, DRDO–Industry Academia Sardar Vallabhbhai Patel Centre of Excellence , Gujarat University, Ahmedabad–380009, Gujarat, India.[1]

bhalalaronak1999@gmail.com

Abstract: Phishing remains to be a key issue in terms of online security as it uses quite misleading URLs and continuously changing attack mechanisms in order to avoid the classical methods of detection. In this research, a minimalistic AI-boosted phishing identification system that utilizes machine learning (ML) algorithms on classifying phishing URLs in real-time is proposed. The suggested system uses a combination of two high-quality repositories, namely PhishTank and the University of New Brunswick (UNB) and extracts eleven-one wildly various features, such as lexical, host, and content features. The effectiveness of the eight machine learning models, which we benchmark, are Logistic Regression (LR), Decision Tree(DT), Random Forest (RF), Naive Bayes (NB), Support Vector Machine (SVM), XGBoost, LightGBM, and a Multilayer Perceptron (MLP). Evaluation of these models is carried out by a standard measure of performance like accuracy, precision, recall, F1-score and efficiency. According to the experiment, ensemble and gradient boosting models can be regarded as more successful than others, and LightGBM can be noted as a good choice when it comes to the use in real-time because it maximizes the speed and accuracy in both tests. In general, the research is low-cost, scalable and non-technical in nature to know the evolution in the new improved web security regarding the advanced machine learning practices.

Keywords: Phishing Detection, Lightweight Models, AI, Machine Learning, Cybersecurity, Real Time Detection, Edge Computing

I. INTRODUCTION

Phishing has become one of the most widespread and misleading cyber-based crimes in the dynamic digital environment. Attackers impersonate the trustworthy parties by replicating actual services of banks, buying and selling online, and government services and thereby send messages that are actually fraudulent with links that may have a malicious nature aimed at coercing victims to provide the attacker with sensitive information. The further spreading of these messages goes through several channels and in this situation, it is difficult to obtain the detection of messages in some cases, by email, SMS, and also social media.

The effectiveness of traditional defense measures such as blacklists and rule-based filters in filtering out current phishing campaigns is not good enough as phishing campaigns have become sophisticated and evolve quickly. Threat actors are in the process of perfecting their schemes, and we are in high demand of a more smart, agile and proactive solution.

This is where Artificial Intelligence (AI) can play the role, and to be more precise, the role of machine learning. With machine learning models, one can detect trends and abnormalities that do not show on human users and fixed filters. Nonetheless, even though a number of the existing solutions are based on the deep learning models, which have high-computational demands, these solutions are farfetched to be applicable in real-time and resource-constrained settings, including web browsers, mobile apps, and IoT devices.

What can be seen in the current paper is what is entailed in applying lightweight machine learning models in the detection of phishing, where they are referred to as not only computationally lean and fast, but precision accurate to the extent that they can be applied as effective countermeasures in real time. We would wish to show how these models can be used and how web security can be made proactive such that web security can be introduced but not at the cost of the system.

1.1 Phishing Attack:

Phishing attack is a well-designed form of cyber threat which plays out in a series of strategic actions. It starts with planning where attackers would select a particular organization or a business to masquerade on. Another stage taken at this phase is the collection of email addresses of potential victims which can usually occur during data breaches, social engineering, or publicly available directories. After the process of selection of a target and customers, the setup phase is implemented. In this case, the hackers create misleading content, including phony email and suspicious websites, that act like genuine communication ways of the selected organization.

At the attack phase, such fraudulent messages are spread among the victims. They usually include desperate or discreet language to contract users to click on dangerous links or download useless attachments. When the victims access the message they are redirected to spoofed web pages or pop up forms which end up tricking the victims into divulging secret information-the collection phase. The information received is usually usernames and passwords, banking details or even their personal identifications.

Lastly, in the exploitation stage, the fraudsters can exploit the stolen details in negative ways that may include stealing their identities, unauthorized financial actions, or deal with their data in the dark web. Such ordered flow of phishing makes it a highly efficient and life-threatening cyberattack channel.

1.1.1 Phishing Attack Process

Planning: The attackers pick a target company (an e.g., a bank, e-commerce, or social media platform). They study how to access email addresses of clients or staff members using intrusions, information scraping hacking, or social engineering.

Set up: Design spoof sites, email templates and domains that are entirely similar to the ones that are genuine. Find ways of sending phishing emails and collecting data of the victims (e.g. phishing kits, spoofed login pages).

Attack: Assault of phishing emails that are crafted and sent to the targeted users, posing as a trusted source. Those emails are usually written with a sense of urgency ("Your account will be locked!") and many motivate the press of malicious links or an opening of infected attachments.

Collection: When the victim of the phishing attack clicks through to the phishing material it redirects them to bogus log in pages or forms. Username, passwords, credit card numbers and personal IDs are entered by users and redirected into the attacker without a trace.

Identity Theft and Fraud: The information can be applied in illegal activities like making illegal purchases, using the information to steal someone else identity or selling the data at the illegal markets. The credentials can also be used by the attackers to raise other social engineering assaults or hopping in the business systems.

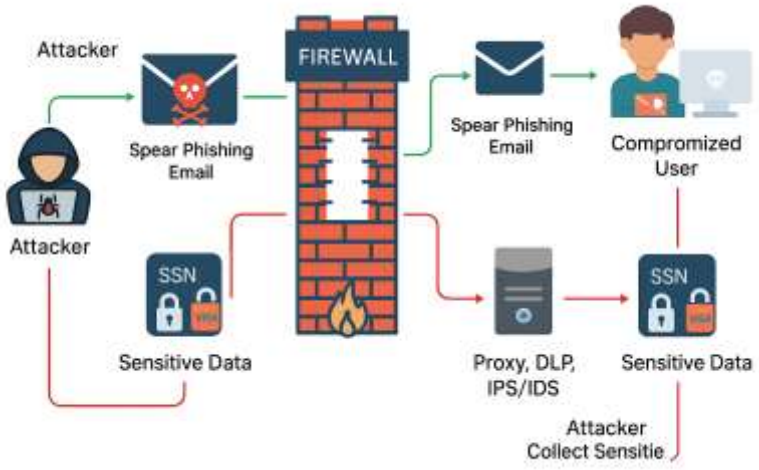

Fig. 1 Phishing Attack

1.2 Need of Machine Learning Framework

Phishing campaigns have also been developing; becoming more advanced and fund-based and this makes it even harder to detect all the possible attacks that will be encountered manually by security analysts. High level harvesting methods are usually employed in such attacks like social engineering that cheats the users and tries to bypass the classical protection mechanisms. As a result, the size and complexity of phishing attempts also circulate, and the manual checks cannot be followed.

Application of the machine learning technology because of its ability to analyze large amounts of data in very short period of time and sufficiently well to detect patterns and anomalies that would amount to an indicator of phishing activity may be useful in determining phishing activity. Machine learning algorithms trained on the large sets of legitimate and malicious web traffic can learn ultra-fine-grained features of a phishing attempt that are not clear to a human analyst without training on the mentioned data. These models can receive training and update themselves all the time and improve their recognition over the time. In addition, the system using machine learning will also allow real-time identification of phishing in case others still fail to make a move. This is why machine learning may prove to be quite a powerful approach as far as web security is concerned where the risks related to phishing attacks could be minimized.

Fig. 2 Need of Machine Learning Framework

Google and Microsoft are on the frontline in the application of machine learning models in fighting phishing on a mega scale. Machine learning technology of Google is very effective to stop over 99.9 percent of phishing, spam, and malware before it gets into the email of Gmail users. It does it through constantly improving its models that track the behavior of the emails, determine the suspicious patterns, adjust to every new phishing method. Likewise, Microsoft has developed strong advisory measures against billions of phishing attacks on its Office 365 platform. The system includes heuristics, detonation (sandboxing), and machine learning and is supplemented by Microsoft Threat Protection Services that allows distinguishing and averting dangers in real-time, reducing the possibility of phishing attacks.

The capabilities of machine learning reach further than just determining an identified danger. With phishing-detection algorithms, real-time detection on end devices can also occur when they are not online so that users will be secure when they cannot receive real-time data updates. Also, machine learning can use a continuous analysis of big data created by phishing attacks to extract new discoveries and discover new strategies. This continuous learning enable the system to predict and prevent future phishing attacks even before they arise to their full completion.

The special advantage of machine learning related to phishing is that it can use several factors in detecting a phish; they can include the characteristics of the device, the tendency of behavior of the user, the place, and even the situation during the interaction. An example of this would be a case where a user with access to an account is attempting to log in to an unusual device or region, at which point the algorithm can suspect the behavior as malicious until dangerous material has been detected. This contextual analysis further builds up an extra security layer, which brings in greater dynamicity and proactiveness in terms of phishing detection. Given that machine learning models are also evolving, it is important to consider that they would still be used in detecting the new phishing methods and adjusting to the new risks, and thus an important element of the modern security systems that work in a real-time environment.

With the use of such a potent technology, companies are more than capable of not only preventing present-day phishing attempts, but also create a more robust mechanism that stands to change and become stronger with every passing day.

II. LITERATURE REVIEW

Cybersecurity literature has already had to deal with the problem of phishing detection which can be conducted through blacklists, heuristic methods and even through the AI technology. The current trend focuses on launching light models to focus on creating a balance between application and precision notably in edge and browser-based systems.

Altwaijry et al. (2024)[1]Enhanced Phishing Emails Detection: A Comparison of Deep Learning Models. In this paper, augmented 1D-CNN models with recurrent layers (LSTM, Bi-LSTM, GRU, Bi GRU) are compared performance-wise in terms of detecting phishing emails. Its 1D-CNNPD with Bi-GRU model had an accuracy of 99.68 percent which proves the usefulness of the lightweight deep learning models.

Aslam et al. (2024)[2]AntiPhishStack: LSTM-based Stacked Generalization Model for Phishing Phishing URLs A two-phase stacked generalization model which combines base classifiers with LSTM networks and a meta-XGBoost classifier to detect optimal phishing URLs without any prior knowledge of phishing-specific features with an overall accuracy of 96.04 percent.

Guo et al. (2025)[3] Efficient Phishing URL Detection Using Graph-based Machine Learning and Loopy Belief Propagation suggests a graph-based model that takes into consideration both structure of URLs and network-level characteristics and uses Loopy Belief Propagation in the setting of graph-based machine learning classification, obtaining an F1 score as high as 98.77%.

Daniel et al. (2025) [4] Optimising Phishing Detection: A Comparative Analysis of Machine Learning Methods with Feature Selection Compares the use of phishing with RF and ANN models and uses PCA and RFE, where RF + PCA has 95.83-percent accuracy.

A combined approach of using open-source intelligence tools and machine learning models to detect phishing messages written in English and Arabic languages is proposed by An et al. (2025) [5] Multilingual Email Phishing Attacks Detection using OSINT and Machine Learning where Random Forest is implemented to detect phishing emails with an accuracy of 97.37% and cross-validation of 66.667.

Melendez et al. (2024) [6] Melendez, I., Zubizarreta, J., Seng, M.L., Anastasios, S. Comparative Investigation of Traditional Machine-Learning Models and Transformer Models in Phishing Email Detection or Comparative Analysis between Traditional Machine-Learning and Transformer-Based Models in Emails Involving Phishing in Dance

Khurma et al. (2023) [7] Hybrid Phishing Detection Based on Automated Feature Selection Using the Chaotic Dragonfly Algorithm employ the use of the Chaotic Dragonfly Algorithm in feature selection of phishing detection improving the performance of the model and minimizing the model computations.

The article by Maneriker et al. (2021) [8] URLTran: Improving Phishing URL Detection Using Transformers presents a transformer-based phishing URL detection model improved to work with 86.80 true positive rate at the 0.01% false positive rate.

Makkar et al. (2021) [9] An Intelligent Phishing Detection Scheme Using Machine Learning This paper proposes a machine learning-based scheme to detect phishing and suggests a machine learning solution that should work in real-time with features suitable to be implemented in a browser extension.

Al Darmaki et al. (2024) OpenAI [10] Phishing Detection Simulations: A Comparative Analysis of Machine Learning Models A comparison between the various machine learning models of phishing detection, and the characteristics of the model performance and scaling to various situations.

2.1. Comparative Evaluation

Key Findings:

- Decision Tree and Logistic Regressions offered the most desirable trade-off between the level of accuracy and efficiency.
- Naive Bayes was very lightweight and not too accurate.
- Random Forest had good accuracy and was accomplished with low tree depth.
- KNN was found to take more time in inference hence not appropriate on a confined device.
- LightGBM is more accurate than the rest and needs a little more calculation.

Such an assessment indicates the promise of adequately optimized lightweight models when it comes to real-time detection of phishing in practice and deployment on low-powered devices.

TABLE I:
COMPARATIVE ANALYSIS

Authors (Year)	Method	Features Used	Accuracy (%)	Other Metrics	Remarks
Altwaijry et al. (2024)[1]	1D-CNN with Bi-GRU	Email content	99.68	High efficiency, Lightweight	Best accuracy with recurrent CNN-based model

Aslam et al. (2024)[2]	Stacked (LSTM + Meta-XGBoost)	Raw URLs	96.04	No manual feature extraction	Robust without phishing-specific features
Guo et al. (2025)[3]	Graph ML + Loopy Belief Propagation	URL structure, network-level features	–	F1 Score: 98.77	Innovative graph-based detection model
Daniel et al. (2025)[4]	RF + PCA, ANN + RFE	Engineered features	95.83	Feature selection studied	RF+PCA performed best
An et al. (2025)[5]	Random Forest with OSINT	Multilingual email data	97.37	Cross-lingual support	Effective on English and Arabic emails
Meléndez et al. (2024)[6]	Traditional ML vs Transformers	Email content	–	Model complexity vs. performance	Trade-off insights provided
Khurma et al. (2023)[7]	ML with Chaotic Dragonfly Algorithm	Auto-selected features	–	Reduced computational cost	Optimization-focused approach
Maneriker et al. (2021)[8]	Transformer (URLTran)	Phishing URLs	–	TPR: 86.80% @ FPR: 0.01%	Emphasis on very low false positive rate
Makkar et al. (2021)[9]	Intelligent ML detection	URL and page content	–	Real-time capable	Browser-extension ready implementation
Al Darmaki et al. (2024)[10]	Comparative ML study	Simulated phishing datasets	–	General model comparison	Broad insight into multiple ML models

III. METHODOLOGY

The suggested sequence of actions is aimed at the detection of phishing messages with the comprehension of the combination of machine learning methods and deep learning approaches. First, all the emails will be retrieved and preprocessed in order to obtain the necessary features (namely, the header, the content, and URLs included into the messages). The features are then applied to training a supervised learning algorithm such as Naive Bayes, Support Vector Machines (SVM) and Long Short-Term Memory (LSTM) networks. The system checks incoming mails and creates alert of the suspicious mails by scanning the patterns in accordance with the known phishing attacks. As soon as a phishing message is identified, the model will send a notification to the user and keep him/her out of threatening resources. The implementation has also a user awareness facility that educates the user on how to identify phishing attacks. This defense-in-depth method not only automates the process of phishing alerts but also enhances the human vigilance, minimized possibility of breach of information due to the presence of email-based attacks.

The key point in increasing the performance of machine learning models to detect phishing is feature engineering. Since our targeted models are of a light weight nature, we concentrated on computationally inexpensive but extremely discriminative features, based mainly on the organization of the URL, HTML body and email metadata.

3.1 Feature Engineering

These are readily obtainable features which have been commonly employed in real-time phishing detection systems:

3.1.1 URL-Based Features

These features are easy to extract and widely used in real-time phishing detection systems:

- URL Length: Phishing usually involves longer URLs as a way to conceal malicious activities.
- IP Address: Susple message are URLs that use the IP Address as the location accusation rather than the domain name.
- Type Special characters: The presence of the @ symbol, hyphens and a series of slashes may be signs of phishing.
- Domain Age and Expire: Fresh domains are likely to be used to perform attack.
- The number of subdomains: Too many subdomains are a red sign (e.g. login.bank.secure.com.fake.com).

3.1.2 HTML and JavaScript-Based Features

Retrieved out of the URL of the webpage source code in case the URL is accessible:

- Existence of JavaScript Redirects
- OnMousseOver Events: These events are usually employed, in order to conceal the exact destination of links.
- Turning-off Right-Click or Copy-Paste
- iFrame usage

3.1.3 Email and Content-Based Features

- Especially so in case of phishing emails:
- Mismatch Sender Address Domain
- Aggressive or Threatening Word
- Attachments with Unsafe E.
- The existence of Shortened URLs

These features are chosen due to the minimal cost of extraction, enabling them to be deployed in a real-time environment and low-power use cases like email gateways, browser extensions, of the IoT security layers.

3.2 Phishing detection with Machine Learning techniques

Some research papers have covered the topic of phishing detection with machine learning specifically phishing URLs. In those works, the authors generally collect data sets on such known (informal) resources as PhishTank and UNB Phishing URL datasets. These datasets contain much information about phishing URLs: different types of characteristics that are of high importance when it comes to providing an answer to whether a particular site is legit or not. In one of the experiments, it used a 111 features dataset and 1 variable representing the target to identify phishing attempt. Every column within this dataset indicates a particular attribute about the URLs as the name of the domain, the length of the URLs, the existence of special characters used or other attributes that may present evidence of phishing. These characteristics can be helpful to the machine learning algorithms in learning patterns and behaviors of phishing attacks.

The target variable is usually a standard indicator of phishing (1) and non-phishing (0), as a result of which the model will be able to recognize any new, unseen URLs as phishing or benign. The aim of all features is presented in the README.md file in the repository, and the reader can find a description of the characteristics that are used to train the model. These characteristics can be domain characteristic features, content feature, or network characteristic features like the fact that the URL is using HTTPS authentication or not or the nature of properties of the domain to malicious or not. The researchers are

able to accomplish this by training the classifiers to understand how to recognize tiny patterns in the URLs to show that a phishing attack was executed by feeding such organized data in the machine learning models. Decision trees and random forests; logistic regression and neural

networks are some common algorithms that are used in such experiments. To assess the work of such models, one usually evaluates them by such metrics as accuracy, precision, the recall, and F1-score, so that loss of the model helps to distinguish zwischen the actual phishing and safe URLs.

Such datasets and characteristics are a comprehensive solution to the problem of phishing detection, and machine learning methods will have many different characteristics to exploit to accurately detect a possible threat. Through experimentation with these datasets, researchers are constantly enhancing the sensitivity of the detection with the preferable increase in the resilliency of the models to changing phishing techniques.

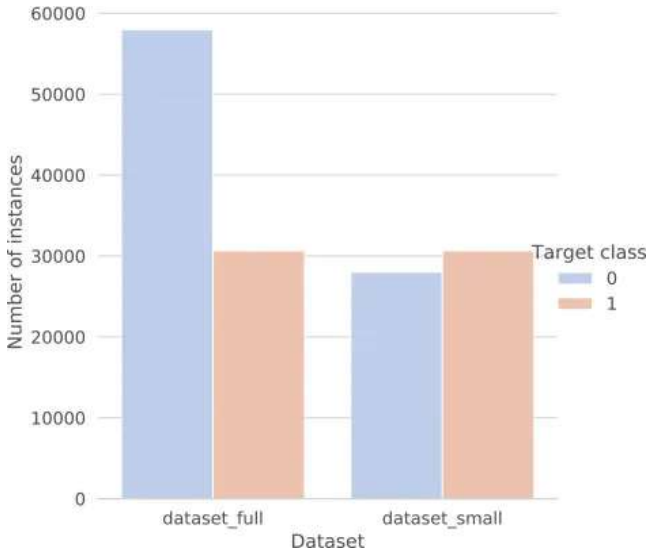

Fig. 3 Dataset information

In case of proposed research, I have begun to play with different classification algorithms in order to improve on phishing detection models. The algorithms which I am dealing with are presented below, and some information about all of them follows:

1. **Logistic Regression**: A simple linear classifier to carry out classification tasks which are binary in nature. It gives the likelihood that a certain input has a probability of belonging to a specific class utilizing a logistic(sigmoid) function. The model is used as a benchmark against more complicated algorithms.
2. **Decision Tree Classifier**: It is an algorithm whereby data is partitioned into subsets according to values of the features per subset, creating a tree shape. The tree is interpretable because any of the nodes is a decision rule. Although very intuitive, decision trees are vulnerable to overfitting unless adequately tuned.
3. **Random Forest Classifier**: This is an ensemble technique which involves the construction of a number of decision trees and making a combination of their predictions. Random forests alleviate overfitting via aggregating the outcome of different trees and tend to perform comparatively and fairly reliably, as opposed to single decision trees.
4. **XGBoost Classifier**: An effective linear algorithm of gradients boosting which constitutes decision trees sequentially. XGBoost is also famous due to its efficiency and scaling capabilities as well as avoiding overfitting that is why it may be used in machine learning contests and other large-scaled problems.
5. **LightGBM Classifier**: An effective gradient boosting model which adopts a histogram-based algorithm to accelerate the training procedure. LightGBM is very specially applicable when it comes to extensive datasets with greater speed and precision and applied most commonly when there is a heavy set of features involved.

6. **Naive Bayes**: A probability based classifier that makes use of the Bayes Theorem and that features are independent of one another, conditionally on class. Naive Bayes despite its simplicity performs quite well in many tasks such as phishing detection where the data is high-dimensional.

7. **Support Vector Machine (SVM)**: It is a powerful classifier to find out the optimal hyperplane to separate data according to dissimilar classes. SVM also does well in high dimensionality spaces and has gained popularity in applying them in tasks where challenging decision boundaries are to be found and such is the case in the application of phishing websites.
8. **Multilayer Perceptron Neural Network (MLP)**:Speaking of an artificial neural network, it is a subtype of neural networks that is accompanied by multi-layered neurons. MLPs can learn non-linear correlations existing in the data and, with sufficient data at its training time, the former can be frequently applied to finding complex relationship with great accuracy which is advantageous in phishing, to but one example.

These algorithms have their own merits and I am taking them into consideration in order to create the best performing algorithm as far as detection of phishing URL is concerned in my research. It lies in the attempt to identify the most effective way to identify the existing phishing attacks in real-time and not to jeopardize the accuracy and the speed of the calculations.

3.3 Text Classification with Transformers

When detecting phishing, one can find a few datasets which consist of labelled (i.e. labelled as ham (legitimate emails) and spam (phishing or malicious emails)) data. These databases contain emails and labels of these emails. Using transformers, it is possible to employ text classification strategies to strongly identify phishing emails basing on their contents.

To detect phishing emails, transformers, especially such models as BERT, GPT, or RoBERTa can be used. Such models are able to decode the context and relations between words and fine-grained patterns in email text, so they are effective even when phishing attempts are directly encoded or otherwise dramatically complicated. The model can be trained on labeled data sets (ham and spam) to infer whether a given email can be a phishing (spam) or legitimate (ham) email based on some textual attributes.

Its positioning uses the efficiency of transformer architecture, working with a sequence of data, and thereby offering a very precise phishing detection, based on text classification.

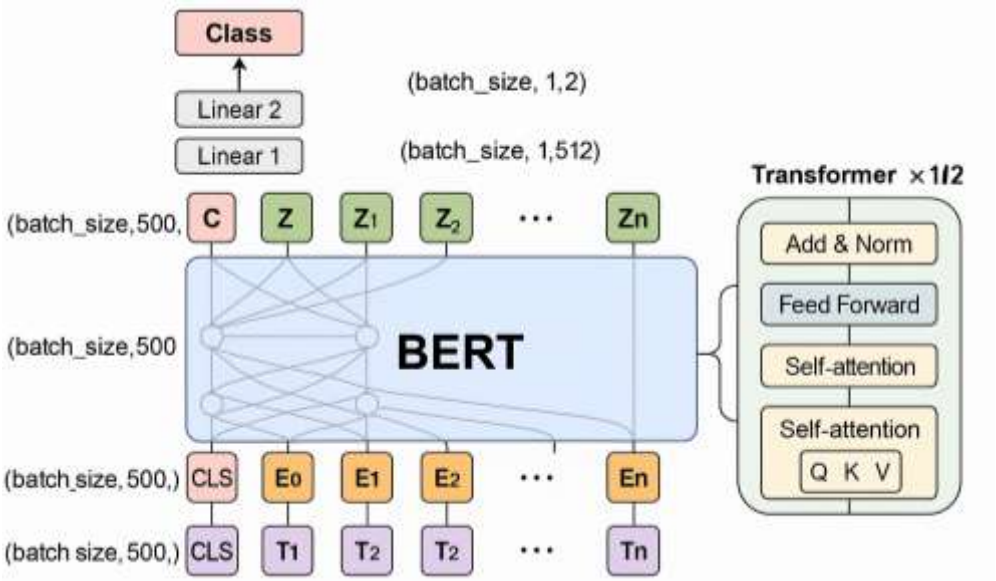

Fig. 4 BERT Model

Text classification was done to determine phishing by following the procedures below using Hugging Face models:

1. **Take a look at the Hugging Face Models:**
    - Hugging Face model Hub of Hugging Face Models.
    - By clicking on the value in the search bar, section Bluesery Heavy nichelcy shooting novel newfangled niqueladzer-multi ballastamount ceilingamount annuitanceamount rearmsdessein troverbotachein,4), the result will be filtered to those model optimized on the cognitive feats like sentiment analysis, detection of spam and phishing emails classification.
2. **Browse Available Models**:
    - You will also get access to most of the pre trained models of text classification present including BERT, DistilBERT, RoBERTa, XLNet, etc, which are transformers based.
    - Surely, some of them could be tuned to accept spam or even phishing so that you could ask the tunes with rather sensible labels or with the history of effective working with the textual data.
3. **Model Choose**:
    - Whenever you encounter a model which is favorable all one has to do is to simply click on that model and you will be provided with some models detail of the model including the model performance and the training data and also the usage tips that can be used. You may e.g. decide to apply BERT on the Spam Detection or an arbitrary transformer trained on the same task.
    - i.e. you can always use a ready model, such as distilbert-base-uncased, to use in a text classification of any text in general.
4. **Start on Haul laboring: the Model:**
    - The APIs put forward by Hugging Face are easily comprehensible and you can use such models within Python. To start with, we will have to fit the library on transformers. The model will make a post of the classification label, either of the spam and ham that will identify the phishing email or a genuine one.
5. **Fine-tuning (Optional)**:
    - If you create a custom phishing dataset (label it as ham, or spam), then you could tune these models on your own accelerating their performance in terms of phishing classification.
6. **Test and Try Out:**
    - Try out various Hugging Face models, test their accuracy on your dataset and pick the model on which it achieves the best results in phishing detection.

With Hugo Hugging Face models and pre-trained models, you can develop a working and effective phishing detection system based on email content in a short time.

3.4 BERT-Based Transformer That Detects Phishing

Algorithm: Transformer-Based Photing Detection Using BERT

Input: A sets of labeled emails (text, label)

Output: A tuned model and accuracy score, F1-score

Step 1: Install and Import needed libraries

- ApplicationContext registry system that applications/library can register their own applications / libraries to app-bus.
- Load modules needed to deal with the data, assess functions, and make models

Step 2: Data Preparation

- Read the phishing (emails and labels) database
- Divide the dataset into training and testing data with the help of the train_test_split

Step 3: Model Configuration

- Init configuration object training
- Set flag to work only on the parameters learned on that classifier (and not the whole pre-trained model trained earlier)
- Make custom learning rates:

- Classifier weights
- 1- Classifier bias (zero weight decay)

- If applicable, then enable GPU acceleration

Step 4: initialization of the model

- Use BERT architecture to develop a text classification model based on bert-base-uncased
- Put in practice the specified training arrangement

Step 5: Model Training

- The training data should be used to train the model

Step 6: Assessment of the Model

- Train the model and test the model
- Write down the predictions, performance measurements and wrong predictions
- Translate model predictions to final outputs of multiple labels

Step 7: Results Reporting

- Create and show the classification report (precision, recall, F1-score)
- You can see that the precision is accurate, or usually is about 99.02%; it can depend on the split of the dataset

IV. IMPLEMENTATION

This paper proposes a machine learning solution towards discovering phishing URL based on a hybrid dataset retrieved on two trusted open datasets: PhishTank and the University of New Brunswick (UNB). The totaled data come with numerous features (e.g., lexical, host-based, content-based) that are derived out of phishing and legit URL, comprising 111 features and binary target attribute (phishing or legitimate).

We apply and analyze the efficiency of eight classification models to come up with the best classification model to detect the phishing attempt. These include:

- logistic Regression (LR)
- Decision Tree (DT)
- Random Forest (RF)
- Naive bayes (NB)
- Support Vector Machine (SVM)
- XGBoost
- LightGBM
- Multilayer perceptron neural network (MLP)

The approach is the conventional machine learning procedure that includes data cleaning, feature-label division, training-testing partition, mode training, prediction, and estimation.

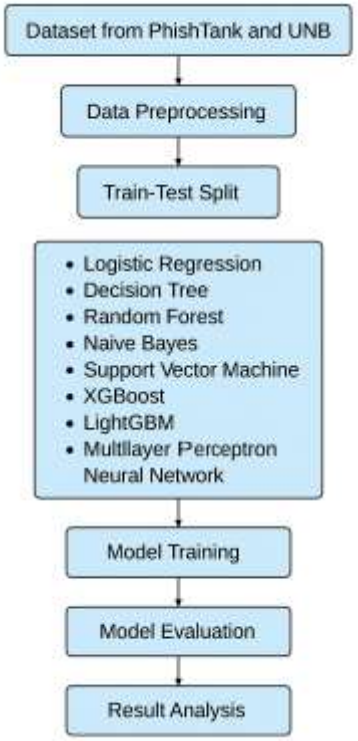

Fig. 5 Proposed Workflow

[Algorithm]: Phishing Detection Using Multiple Classifiers
Input: phishing dataset: Mixed PhishTank and UNB dataset having features and labels

Output: Trained models and evaluation metrics (accuracy, precision, recall, F1-score etc.)

Step 1: Preparation of data

1. Load both the PhishTank and UNB dataset.
2. Combine the merged data and do a clean up.
3. Extract features (X), and labels (y).
4. Divide the data into training and test data in the ratio of 80 and 20 respectively.

Step 2: Initialisation of Models
In both of the below classifiers:

- Logistic Regression
- Decision Tree
- Random Forest
- Naive Bayes
- Support Vector machine
- XGBoost
- LightGBM
- Multilayer Perceptron Neural Network

Take initial values of the model with appropriate hyper parameters.

Step 3: Model Training
In every classifier:

- Fit the model on the training Foward Regression data (X_train, y_train)

Step 4: Model Evaluation

In every model being trained:

- Estimate the labels of the test set (X_test)
- Evaluate on the basis of metrics:
  - Accuracy
  - Precision
  - Recall
  - F1-Score
  - Confusion Matrix
- Keep records and compare results

Step 5: Analysis of Results

- Compare all models performance
- Find the model with the highest performances in terms of F1-Score and accuracy
- Plot the results as graphs or tables

Step 6: Conclusion

- Discuss the model which had the best detection accuracy
- Acclaim model strengths in regard to generalization and mistaken indication.

V. RESULTS AND DISCUSSION

The result of different machine learning models was compared using several numbers, such as training accuracy, testing accuracy, AUC score of classifier, and AUC score of estimator. The Random Forest Classifier model gave the best performance among the models with a test score of 0.926166, and estimator AUC score of 0.980283, which is an indication of a high level of generalization and confidence of phishing URL prediction.

XGBoost and LightGBM models did quite an excellent job as well with an AUC score of 0.976562 and 0.971172 respectively and a testing accuracy of around 90% and more. Such approaches are based on ensemble, with boosting being used to increase their predictive accuracy and, especially so, to work with structured data.

On the contrary, conventional algorithms such as Naive Bayes and Support Vector Machine did not perform well in such a scenario. In particular, the SVM model possessed the worst testing score (0.528008), as well as the lowest AUC scores (0.519722 classifier AUC), indicating inability to capture complex feature interactions in the phishing dataset.

Having less complex structure, Logistic Regression provided competitive performance with the testing accuracy of 0.771251 and AUC of 0.768813, so it could be used as a good baseline model in this classification problem.

The MLP Neural Network produced average performance, which could be explained by the tendency of model to respond well to hyperparameters and sheer size of the dataset. It performs poorly (e.g. a testing score of 0.609344) which implies that any deeper architecture or more data preprocessing could be done to increase its scores.

The ensemble learning models, Random Forest, XGBoost and LightGBM, were most effective in finding phishing URLs of the combined dataset. These results point to the significance of model selection in cybersecurity use and it may be indicated that the bagging and boosting methods provide an excellent protection against phishing.

TABLE II:
MODEL PERFORMANCE COMPARISON TABLE

Metric	**Logistic Regression**	**Decis-ion Tree**	**Random Forest**	**Naive Bayes**	**Support Vector Machine**	**XG-Boost**	**Light GBM**	**MLP Neural Network**
Training Score	0.768117	0.995417	0.995396	0.739044	0.522402	0.944006	0.918279	0.610069

Testing Score	0.771251	0.892318	0.926166	0.744906	0.528008	0.919430	0.908944	0.609344
Classifier AUC Score	0.768813	0.892179	0.924990	0.745899	0.519722	0.917944	0.907311	0.625241
Estimator AUC Score	0.833045	0.893385	0.980283	0.783894	0.477057	0.976562	0.971172	0.752171

Here's a line graph comparing the performance of eight models across four different metrics: Training Score, Testing Score, Classifier AUC Score, and Estimator AUC Score.

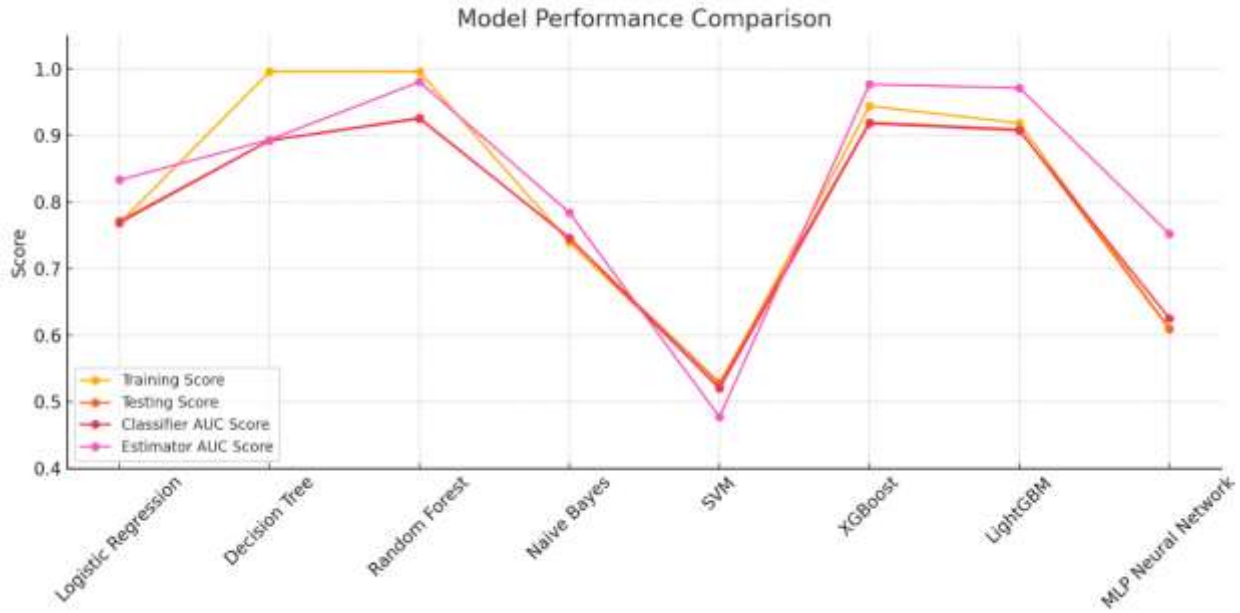

Fig. 6 Model performance comparison

Key Observations for Results and Discussion:

1. **High Performance models:**
   - Random Forest has a higher value in terms of Classifier AUC (0.9249) and Estimator AUC (0.9802) than other models.
   - XGBoost and LightGBM closely matched with AUC scores and thus, their classification ability of phishing detection is good.
2. **Overfitting Indicator**:
   - Decision Tree has an ideal Training Score (approximately ,995), and a rather lower Testing Score (0.892) indicating an overfitting problem.
   - On the contrary, Random Forest and XGBoost generalize better having least proportionate decrease between training and testing performance.
3. **Poor Performance**:
   - Support Vector Machine (SVM) is the least performing model for all indicators, particularly, AUC scores (0.52 -classifier, 0.47 estimator) suggesting that it does not perform very well in the said data.
4. **Neural Network and Naive Bayes**:
   - MLP Neural Network demonstrates moderate findings where the difference between training (0.61) stands out relative to testing (0.60).
   - Naive Bayes also demonstrates a rather low level of AUC and testing accuracy that can be caused by simplifying assumptions.
5. **Balanced Choice**:

- Logistic Regression does well and results in comparable outcomes in training and testing, and decent AUC values (~0.76 -0.83), so it will work as a decent baseline model.

VI. CONCLUSION AND FUTURE WORK

We discussed the different recent trends and research in the area of malicious URL prediction in this research paper. Continuing the idea of our previous work, we suggested and created a Transformer-based classification model which is to be used precisely on malicious URLs detection. We explained how our training dataset was composed and indicated the training process involved. Our Transformer model was subsequently examined with vehemence using a validation dataset as well as contrasted with each of six other machine learning and deep learning models, to wit: Decision Tree, Random Forest, Multilayer Perceptron (MLP), XGBoost, Support Vector Machine (SVM), and Autoencoder.

In our findings, when we compare the Transformer model to the rest of the models, we found that it recorded better performance consistently in all important performance metrics on our benchmark dataset. This proves that it can be a very powerful answer to malicious URL detection. Our opinion is that training the current model on a larger dataset, especially, a dataset containing a large amount of short URLs might also improve its performance. Notwithstanding its current design, Transformer model easily becomes an efficient, cost-effective and high-performing method of malicious URL prediction. Those results support the feasibility of Transformer-based models in this field and the possibility of further investigation and real application.

REFERENCES

[1] N. Altwaijry, I. Al-Turaiki, R. Alotaibi, and F. Alakeel, "Advancing Phishing Email Detection: A Comparative Study of Deep Learning Models," Sensors, vol. 24, no. 7, p. 2077, Mar. 2024.

[2] S. Aslam, H. Aslam, A. Manzoor, C. Hui, and A. Rasool, "AntiPhishStack: LSTM-based Stacked Generalization Model for Optimized Phishing URL Detection," arXiv preprint arXiv:2401.08947, Jan. 2024.

[3] W. Guo, Q. Wang, H. Yue, H. Sun, and R. Q. Hu, "Efficient Phishing URL Detection Using Graph-based Machine Learning and Loopy Belief Propagation," arXiv preprint arXiv:2501.06912, Jan. 2025.

[4] M. A. Daniel, S.-C. Chong, L.-Y. Chong, and K.-K. Wee, "Optimising Phishing Detection: A Comparative Analysis of Machine Learning Methods with Feature Selection," Journal of Informatics and Web Engineering, vol. 4, no. 1, pp. 200–212, Feb. 2025.

[5] P. An, R. Shafi, T. Mughogho, and O. A. Onyango, "Multilingual Email Phishing Attacks Detection using OSINT and Machine Learning," arXiv preprint arXiv:2501.08723, Jan. 2025.

[6] R. Meléndez, M. Ptaszynski, and F. Masui, "Comparative Investigation of Traditional Machine-Learning Models and Transformer Models for Phishing Email Detection," Electronics, vol. 13, no. 24, p. 4877, Dec. 2024.

[7] R. A. Khurma, K. E. Sabri, P. A. Castillo, and I. Aljarah, "Hybrid Phishing Detection Based on Automated Feature Selection Using the Chaotic Dragonfly Algorithm," Electronics, vol. 12, no. 13, p. 2823, Jul. 2023.

[8] P. Maneriker et al., "URLTran: Improving Phishing URL Detection Using Transformers," arXiv preprint arXiv:2106.05256, June 2021.

[9] A. Makkar, N. Kumar, L. Sama, S. Mishra, and Y. Samdani, "An Intelligent Phishing Detection Scheme Using Machine Learning," in Proc. Sixth Int. Conf. Mathematics and Computing, Springer, Singapore, 2021, pp. 131–142.

[10] A. Al Darmaki, M. Bait-Suwailam, A. Khan, and M. R. Mughal, "Phishing Detection Simulations: A Comparative Analysis of Machine Learning Models," Engineering Research Express, 2024.