International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Phishing Website Detection Using Advanced Machine Learning Techniques

Dr. Nitin N. Sakhare[1,*], Jyoti L. Bangare[2], Dr. Radhika G. Purandare[3], Disha S. Wankhede[4], Pooja Dehankar[5]

Submitted: 20/11/2023 **Revised**: 27/12/2023 **Accepted**: 08/01/2024

Abstract*:* In the contemporary digital landscape, the escalating threat of phishing attacks necessitates innovative solutions for timely detection and mitigation. This paper presents a pioneering endeavour at the intersection of Artificial Intelligence (AI) and Machine Learning (ML) to combat phishing attempts. Employing a multifaceted approach, this research work integrates XGBoost, LightGBM, Naïve Bayes and CatBoost algorithms, alongside a Graph Neural Network (GNN), to meticulously analyse URL structures, content patterns, and user behaviour. Features such as URL length, dots, slashes, numbers, and special characters are extracted for comprehensive model training. Real-time monitoring ensures the continual adaptation of the system to emerging phishing tactics, enhancing its efficacy in proactively safeguarding users and organizations from the dynamic and evolving realm of cyber threats. This research encapsulates a comprehensive exploration of diverse machine learning methodologies to fortify online security against the pervasive threat of phishing.

Keywords: *Phishing Detection, Artificial Intelligence (AI), URL Analysis, XGBoost, LightGBM, Naïve Bayes, CatBoost, Graph Neural Network (GNN), Feature Extraction, Real-time Monitoring*

I. Introduction

The exponential growth of online activities in recent years has brought about an alarming rise in cyber threats, with phishing attacks standing out as a pervasive and cunning menace. Phishing, the deceptive practice of masquerading as a trustworthy entity to obtain sensitive information, poses a substantial risk to both individuals and organizations. In response to this escalating challenge, research work presented here is a sophisticated initiative that leverages the power of Artificial Intelligence (AI) and Machine Learning (ML) to proactively detect and counter phishing attempts. As cybercriminals continuously refine their tactics, the research is conducted to stay ahead of the curve, employing a diverse set of machine learning algorithms, including XGBoost, LightGBM, Naïve Bayes, CatBoost, as well as innovative Graph Neural Networks (GNN) to scrutinize URL structures and user behavior. This research explores the intricacies of feature extraction, real-time monitoring, and continuous learning, offering a robust and adaptive solution to fortify online security.

The key components of the Phishing Website Detection using AI, elucidating the significance of each algorithm and the comprehensive methodologies employed. The integration of traditional machine learning models with state-of-the-art graph-based techniques enhances our system's ability to discern intricate patterns, providing a formidable defence against the dynamic landscape of phishing threats. By bridging the gap between advanced AI technologies and the imperative need for heightened cybersecurity, Phishing Website Detection using AI endeavours to usher in a new era of online safety, shielding users from the ever-evolving tactics employed by cyber adversaries.

II. Literature Survey

The authors of [1] compared various studies identifying phishing assaults for each AI methodology and looked at the benefits and drawbacks of various methodologies. The

1,3,4BRACT's Vishwakarma Institute of Information Technology, Pune
2MKSSS's Cummins College of Engineering, Pune
5Assistant Professor
School of Engineering, Ajeenkya D. Y. Patil University, Pune
pooja.dehankar@adypu.edu.in
*nitin.sakhare@viit.ac.in**

communication medium, target devices, attack method, and countermeasures are the four facets of a phishing attempt that are covered in this survey. To identify a phishing assault from these, machine learning and deep learning techniques are primarily used. Other popular classification techniques include RF, SVM, C4.5, DT, PCA, and k-NN. A more scalable and robust strategy, including smart plugin solutions, is still being investigated to tag or label whether the website is real or pointing towards a phishing assault, even though these methods are the most beneficial and successful for identifying phishing attacks.

The CIC-Bell-DNS 2021 Dataset, which includes 400,000 benign and 13,011 malicious samples from a million benign and 51,453 known-malicious domains, was used by the authors in [2]. The dataset correctly depicts actual occurrences, including both typical, beneficial traffic and many forms of harmful domains, including spam, phishing, and malware. This study strengthens cybersecurity defences against emerging threats by accurately classifying hostile sites. Each method showed both its advantages and disadvantages in identifying rogue DNS. The Canadian Institute of Cyber Security provided the CIC-Bell-DNS 2021 dataset, which underwent pre-processing using principal component analysis to reduce the risk of over-fitting. Several supervised ML systems were then trained using the pre-processed dataset. Analysis was done to assess the models' accuracy and other indicators of performance. Results show that the proposed model outperformed both existing models and the state-of-the-art model, pointing to a promising avenue for improving the detection of malicious DNS; however, there is still room for improvement, such as combining multiple techniques or creating hybrid approaches that take advantage of the advantages of various methods, which would help the field advance. Another area where there is room for improvement is the addition of attribute-based static features like opcode or other attribute-combination features, as well as more stateful/stateless features.

In [3], the authors investigated how current phishing feature datasets may be effectively incorporated into a countermeasure. This was accomplished by choosing their suggested feature vector based on the ranking of several feature categories from previously published literature. It was suggested that the effectiveness of anti-phishing strategies be increased using an improved machine learning-based predictive model. The feature selection module of the predictive model was employed to create an efficient feature vector. The predictive model used in this study takes use of the adequate integration of the top relevant phishing "fingerprints," presenting the benefits of phishing detection based on readily available high-ranking feature vectors. However, there is still room for improvement. For example, investigating the use of their approach on proprietary middleware like SOAP, investigating the use of their design as mobile apps for smartphone-based phishing attacks, examining the suitability of their design in the emerging IoT-based phishing attacks, and finally investigating the contributory influence or complementing effects of various features using more intensive study on appropriate theoretical framework were all things to consider.

The authors of the research [4] provide insight into the various phishing website detection methods, the data sets utilised, and the algorithmic performance evaluations. In this study, 537 research articles from five electronic libraries were analysed; 238 articles remained after applying inclusion-exclusion criteria. 80 studies were left after the third elimination criterion. To steer the study in the right direction, a study of these 80 publications was conducted using research questions. These research questions will aid in determining which technique, dataset, and algorithm were most frequently utilised in the literature as well as which algorithm or technique performed the best in terms of accuracy. The accuracy of the CNN algorithm was the greatest among all the experiments in this study, at 99.98%, even though the most popular method utilised here was Random Forest among the more conventional Machine Learning algorithms. No matter the data set or features that were retrieved for the prediction analysis, it was the same.

The study in [5] discusses improved detection methods that make use of machine learning technologies to identify phishing URLs. It is made up of different Machine Learning techniques that have been used to identify phishing URLs. A different ML model that had been successfully detecting phishing URLs using Logistic Regression with a precision of over 97% was used to derive and modify the best fitting strategy. The model must still be designed with the Random Forest algorithm and

the blacklist method in order to create a scalable web-based phishing detection system.

In order to effectively detect phishing sites, particularly in the real web environment, where high efficiency and performance as well as extremely low false alarm rates are required, the authors of this paper proposed a comprehensive and interpretable CASE feature framework and designed a multistage phishing detection model. The proposed method works well in actual phishing discovery, demonstrating its high practicability in the real world, and delivers superior detection results under the assumption of drastically reducing the execution time. To effectively assist the validation of novel detection models, aspects like expanding a dataset to include more brands and languages are required. Further research into the model layer fusion can also be used to extend the framework [6].

[7] provides a thorough analysis and overview of the most recent BEC phishing detection methods. To give readers a rudimentary understanding of the BEC phishing assaults, the findings included a condensed version of a few chosen publications. But the difficulties we currently face and the future directions of our study in BEC phishing detection are based on ML. To create an efficient and optimised BEC phishing detection system, more research is required on dynamic feature selection, building real-world datasets, merging NLP with deep learning, and combining ML with XAI.

In [8], the authors compared the potency of ML classification models in phishing domain detection. The result using the Gradient Boosting-based model in combination with Random Forest demonstrated higher performance when compared to the other strategies and is consistent with previously published solutions. The accuracy rates of the Gradient Boost and Random Forest models both match those reported in the published literature, displaying excellent performance. This extraordinary potential makes these models strong contenders for practical implementations in phishing detection scenarios, strengthening cybersecurity defences and protecting users from the risks posed by phishing attempts.

To recognise phishing websites, the authors of [9] use Adaboost and Multi Boosting ensemble algorithms. The main goal was to make these phishing detection algorithms more effective. Given the offered model that the authors explored, it was possible to identify phishing pages with an accuracy of 97.61 when ensemble models were implemented. Deep learning is necessary for the detection of phishing websites, though.

In order to provide a taxonomy of deep learning algorithms for phishing detection, the study carried out a systematic literature review (SLR) by looking at 81 chosen publications. The SLR was conducted in the study using Kitchenham's four-phase approach, which included research questions, a search process, article selection, and data synthesis. To locate pertinent publications published between 2018 and 2021, an automatic search strategy was employed, executing a Boolean search string on several database sites, including Web of Science, IEEEXplore, Springer Link, Science Direct, and Google Scholar. The study offers a thorough analysis of the benefits and drawbacks of the most recent deep learning methods for phishing detection. The paper also addresses the problems that deep learning has when it comes to phishing detection and suggests future research avenues to address these problems. An empirical analysis is carried out to assess the effectiveness of several deep learning approaches in a real-world setting, emphasising the relevant problems that will spur future research. [10]

An overview of deep learning techniques and their use in several cybersecurity fields, such as malware analysis, network security, intrusion detection, and phishing detection, is given in the study [11]. It talks about how deep learning may improve cybersecurity measures by leveraging its strengths in feature extraction, pattern recognition, and anomaly detection. To counter increasing cybersecurity threats like malware, phishing, DDoS attacks, and advanced persistent threats (APTs), the study report emphasises the need for sophisticated solutions. It also discusses the shortcomings of these algorithms and highlights the significance of deep learning in bolstering cybersecurity measures. The study makes recommendations for future lines of inquiry to develop deep learning in cybersecurity. Overall, the study offers a thorough overview of the literature on deep learning algorithms and how they are used in cybersecurity, stressing the benefits, drawbacks, and potential paths for future research.

The use of machine learning algorithms to identify phishing domains is covered in the study [12]. It contrasts four models that were created with the help of random forest (RF), decision trees (DT), support vector machines (SVMs), and artificial neural networks (ANNs). As a benchmark, the UCI

phishing domains dataset is used to assess the models. The results demonstrate that, out of the four models, the random forest technique is the most accurate and performs better than other solutions found in the literature. ANNs, SVMs, DTs, and RF are examples of common machine learning classification approaches that have been shown effective in phishing domain identification.

In [13], authors have expanded upon the findings of previous studies that have been documented in the literature. Comparing the classifiers employed in this paper to those reported in the literature using the same datasets, they performed similarly or even better in some cases. The paper's findings show how learning machines can be used to identify and categorise phishing emails. In subsequent study, the authors intend to compare accuracy rates using additional machine learning techniques. To identify the set of features that consistently yield the best accuracy across all classifiers, they also intend to perform a complete feature ranking and selection on the same dataset. Overall, by adding new variables and demonstrating the effectiveness of their classifiers in identifying and categorising phishing emails, the work expands upon prior research in the field. The authors also point out the possibility for additional study in terms of investigating additional machine learning methods and performing feature selection and ranking.

In [14], the authors described modelling a Dense network model-based effective URL phishing detection method. Using a dense forward-backwards long short-term memory model, 98.5% accuracy in phishing attack detection is attained. The model works better than other systems and is not dependent on outside services. The COVID-19 pandemic's rise in phishing attacks is linked to people's increased reliance on online services. Local characteristics in traffic data are captured by the model's convolutional layer. The system is assessed using performance metrics like recall, accuracy, F-measure, and precision. The suggested model exhibits excellent accuracy and precision without requiring outside services. Additionally, the system looks for character-level similarities when identifying phoney phishing URLs.

The authors pointed out in their work that the goal of this research [15] is to develop a machine learning model that can identify phishing websites using a variety of methods. The use of the internet has increased, which has increased criminality, especially phishing attacks. A dataset of 3000 URLs—both benign and malicious—is used in the study. Using the Light GBM technique, the machine learning model outperformed the random forest and decision tree algorithms in terms of accuracy. A graph illustrates the relative relevance of the various features in the model. The various algorithms' testing and training accuracies are also given.

The study [16] focuses on analysing common characteristics displayed by phishing websites and creating a model to identify them. The dataset was used to train several models, including the Max Vote Classifier, K Nearest Neighbours, Decision Tree Classifier, Random Forest Classifier, and Logistic Regression. At 97.73% accuracy, the Max Vote Classifier comprising Random Forest, Decision Tree, and Artificial Neural Network achieved the maximum accuracy. The study suggests developing an online application that allows users to input a website URL and uses a trained model to determine if the website is phishing or not. The article also covers characteristics such as whether a website link contains the"" mark, how many links go to the page, how popular the website is according to Alexa rank, and how much information is duplicated. All things considered, the study offers insights into the analysis and detection of phishing websites through a variety of models and suggests a useful application for everyday use.

In [17], Phonological analysis, and many classifiers, including Decision Tree, K-Nearest Neighbour, Random Forest, Gaussian Naïve Bayes, and Logistic Regression, were used in a study on the classification of phishing URLs. The study included feature engineering, dataset randomization, and regular expressions and lexical analysis-based extraction. The results demonstrated that different classifiers produced similar results, with dataset randomization improving classifier accuracy. The classifiers with the highest accuracies were Gaussian Naïve Bayes and Random Forest, with the latter obtaining the maximum AUC (area under the curve) of.991. Despite the excellent performance of all classifiers, Naïve Bayes was shown to be more appropriate for this job. Adding more features and continuously training with updated datasets were suggested as ways to enhance accuracy even more. Since it can be difficult to find and maintain phishing websites, content-based classifiers were not trained using content-based features in this study.

The study in [18] contrasts deep learning and machine learning methods for using URL analysis to detect phishing URLs. The technique employs URLs from the login page in both phishing and genuine classes in order to generate a more realistic dataset. The study examines various techniques for feature extraction, such as statistical features utilising Term Frequency-Inverse Document Frequency (TF-IDF) in conjunction with character N-gram and handcrafted features suggested by the authors. Additionally, it uses a Gated Recurrent Neural Network (GRU) and CNN models for deep learning methods. The introduced login URL dataset exhibits 96.50% accuracy when the Logistic Regression model is paired with TF-IDF feature extraction, according to the results.

Currently used methods for identifying and stopping harmful URLs mostly involve blacklisting and supervised machine learning. The first line of defence against dangerous websites is blacklisting, such as Google Safe Browsing, albeit this method is not 100% successful. Supervised machine learning techniques have demonstrated efficacy in classifying URLs as either benign or harmful. Additional strategies include DNS traffic analysis, associative rule-based mining techniques, and visual similarity signatures. To identify phishing attempts that can elude first detection sensors, layered phishing detection techniques have been put forth. For the purpose of phishing detection, some earlier studies merged neural networks and reinforcement learning with heuristic-based detection systems [19].

In order to overcome this flaw, the SPEDAS model that is presented in [20] looks for phishing emails that contain similar-sounding terms. The method for creating keywords that sound alike is also covered in the study. It uses three models that were carried over from previous research. For instance, by adding an extra character "o" to the string, the word "account" can be changed into a term that sounds similar, producing the phrase "accoount." addresses the use of datasets from the UCI Machine Learning Repository to illustrate the application of Random Forest and SVM classification techniques for phishing detection. While the SVM classification algorithm obtains an accuracy of 92.62%, the Random Forest approach achieves 95.1% accuracy.

The current machine learning techniques for identifying phishing websites are described in this study [21]. With implementation accuracies of 97.369%, 97.451%, and 97.259%, respectively, the paper describes the enhanced Random Forest classification method, SVM classification algorithm, and Neural Network with backpropagation classification methods. In this case, the SVM classification algorithm was selected as the final classifier method for the classification of websites as phishing or authentic because it provided superior accuracy than the Random Forest and Neural Network classification algorithms.

The study in [22] offers a survey of machine learning-based phishing website detection. Numerous phishing detection strategies, such as blacklist, heuristic, content analysis, and machine learning approaches, have been researched. The authors investigated classification methods on 30 features of phishing websites using Extreme Learning Machine (ELM) and obtained an accuracy of 95.34% by utilising 6 distinct activation functions. Using machine learning models, the authors proposed the identification of phishing assaults; Support Vector Machine (SVM) produced the greatest results, while Naive Bayes and Artificial Neural Network produced results with an accuracy of 97.98%.employing ensemble approaches like stacking, bagging, and boosting together with Naive Bayes, Decision Tree, and Random Forest, the authors proposed employing machine learning for phishing detection with characteristics taken from the URL only, reaching an accuracy of 97.08%.

The use of machine learning algorithms to identify phishing attacks—a popular kind of cyberattack used to acquire user data—is covered in [23]. The authors point out that to address security-related problems, machine learning and deep learning approaches have been applied extensively in recent years. The study uses a variety of machine learning techniques, including gradient boosting classifier, AdaBoost, logistic regression, decision tree, random forest, and classifier for decision trees, to detect phishing attempts. The authors demonstrate that their fusion classifier with two priority algorithms outperforms earlier models by achieving an accuracy of 97% when compared to earlier efforts. The UCI machine learning repository provided the dataset utilised in the studies.

The study in [24] presents a thorough analysis of the current state of the art in the detection of phishing websites, outlining the key issues and conclusions. List-based, similarity-based, and machine learning-based are the three primary categories into which the

survey divides the detection techniques. The study outlines the detection techniques put out in the literature for each category as well as the datasets taken into account for evaluation. The survey also identifies a few areas of research in the field of phishing website identification that still require attention.

In the past, phishing detection research has concentrated on identifying the differences and similarities between phishing and legitimate emails by extracting highly distinctive elements from the email text and metadata. Numerous methods have been employed, such as information gain measurements, Principal Component Analysis (PCA), Latent Semantic Analysis (LSA), Mutual Information, and Chi-Square statistics. When compared to various state-of-the-art studies, the suggested approaches showed optimal performances with reduced feature sets based just on the text. Future work will be required to implement methods based on deep learning, language models, and transformers to detect phishing [25].

III. Introduction

This paper holds a broad scope in the realm of cybersecurity, aiming to fortify defenses against the growing menace of phishing attacks. By integrating new age machine learning models such as XGBoost, LightGBM, Graph Neural Networks, Catboost, and Naive Bayes Classifier. The research work encompasses a diverse set of methodologies to scrutinize URL structures, content patterns, and user behaviour. Its comprehensive feature extraction techniques delve into the intricacies of URLs, including length, special characters, and numerical components, enhancing the system's robustness in detecting sophisticated phishing attempts.

Beyond static analyses, this work extends its scope to real-time monitoring, ensuring a vigilant stance against emerging phishing threats. Its adaptive nature allows the system to dynamically evolve and refine its detection capabilities in response to the ever-changing tactics employed by cyber adversaries. This adaptability is a crucial aspect of the scope, recognizing the dynamic cyber threat landscape. Moreover, Phishing Website Detection using AI's scope isn't limited to technical aspects alone; it extends to user and organizational protection. The research aims to contribute to a heightened cybersecurity posture for both individual users and entities navigating the complexities of the online environment. By bridging the interdisciplinary realms of AI, ML, and cybersecurity, Phishing Website Detection using AI aspires to not only detect and prevent phishing attacks but also to contribute to educational initiatives, fostering awareness about safe online practices and providing insights into evolving cyber threats. In summary, the scope of Phishing Website Detection using AI encapsulates a holistic and adaptive approach to address the multifaceted challenges posed by phishing attacks in the contemporary digital landscape.

IV. Proposed Methodology

The surge in phishing attacks poses a critical challenge to the security of online users and organizations. As cybercriminals continually refine their tactics to mimic legitimate websites, the ability to effectively detect and counteract these deceptive practices becomes paramount. Current cybersecurity measures often fall short in providing comprehensive protection, necessitating an advanced solution that combines the strengths of Artificial Intelligence (AI) and Machine Learning (ML). The problem addressed by this research work is the need for a robust and adaptive system capable of accurately identifying phishing domains by analyzing URL structures, content patterns, and user behavior. This research seeks to fill the existing gap in cybersecurity defenses, providing a proactive and sophisticated solution to safeguard users from the evolving and increasingly sophisticated landscape of phishing threats.

The architecture devised for detecting phishing websites adopts a comprehensive approach, integrating conventional machine learning models—namely XGBoost, LightGBM, and a referenced but inactive Random Forest classifier—alongside a Graph Neural Network (GNN). The methodology follows delineated phases for effective implementation. Figure 2. shows us the data flow diagram. Primarily, the conventional machine learning segment commences by loading and preprocessing a dataset containing URLs and corresponding labels. Various features are extracted from the URLs, encompassing attributes such as URL length, punctuation utilization, and structural characteristics. Subsequently, the dataset undergoes splitting for training and testing, followed by TF-IDF vectorization to convert textual information into numerical features. While the code includes a commented-out section for a Random Forest classifier, it remains inactive within the system.

Secondly, the Graph Neural Network (GNN) module selectively samples a subset of the dataset for computational efficiency. It constructs a graph representation based on the disparity in URL lengths, defining node features and establishing the architecture of the GNN model. This GNN model is trained using labeled data, employing the Adam optimizer and cross-entropy loss, thereafter, undergoing evaluation on a distinct test set with performance metrics computed for thorough assessment. Furthermore, the system enables real-time inference on new URLs by leveraging XGBoost, LightGBM, GNN, Catboost, and Naive Bayes Models. It incorporates ensemble learning functionalities, amalgamating predictions derived from both the traditional machine learning models and the GNN model. Lastly, the architecture accounts for model persistence by specifically saving the trained GNN model to a designated file ('gnn_model.pth') for subsequent retrieval during inference.

Essentially, this architectural design shown in Figure 1 harmonizes the capabilities of contemporary machine learning models and a Graph Neural Network to fortify the identification of phishing websites. Beyond model training and evaluation, it accommodates real-time predictions, harnessing the diverse strengths of these models to augment system efficacy.

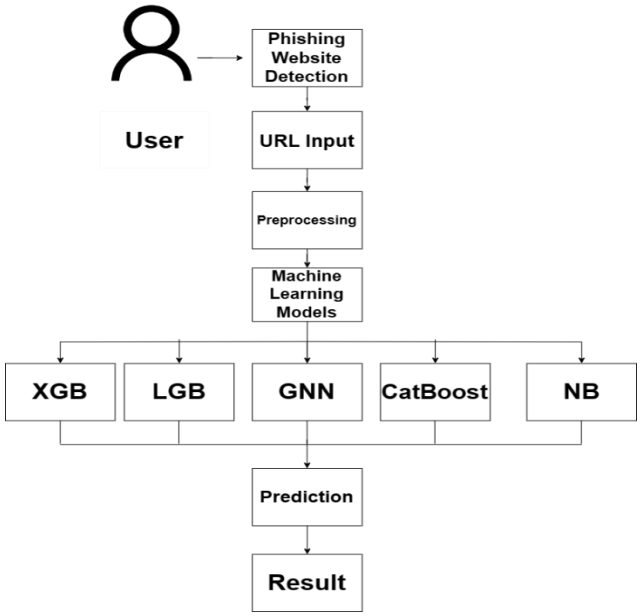

Fig 1. Architectural Diagram of URL Phishing Detection

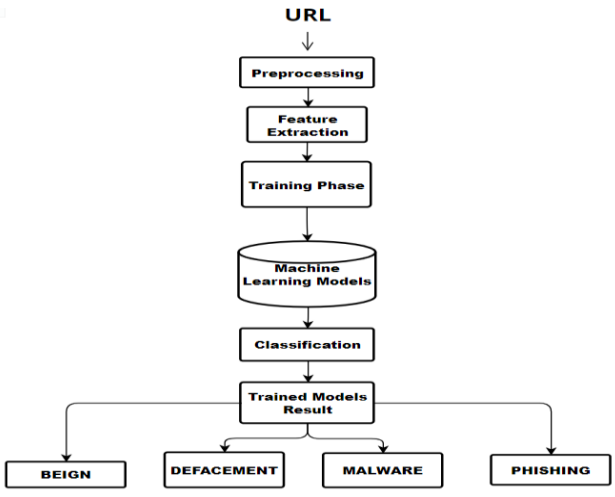

Fig 2. Data Flow Diagram

Overall methodology encompasses a series of critical steps to effectively identify and categorize URLs as either benign or potentially malicious. It commences with the acquisition of a dataset from a CSV file, containing vital information about URLs, such as their structure, alongside a classification denoting whether the URL is associated with phishing or considered benign. To ensure the data is in a suitable format for analysis, the dataset undergoes a thorough examination using the info() method, allowing for essential preprocessing steps to be implemented. This preliminary stage is crucial in understanding the dataset's structure and contents, facilitating any necessary data cleaning or normalization processes. The dataset is then prepared for machine learning model training. The 'url' column, carrying the URL information, is separated as the feature variable (X), while the 'type' column, denoting the classification of the URLs, is segregated as the label (y). The labels are encoded into numerical values using Label Encoding, a pivotal preprocessing step required for training machine learning models effectively. Subsequently, the dataset is split into distinct training and testing sets using the train_test_split function. This division is essential as it allows for the training of machine learning models on one subset while enabling the evaluation of their performance on an unseen subset, ensuring the models' robustness and generalizability. The prcoess then delves into the transformation of URLs into numerical vectors using the Term Frequency-Inverse Document Frequency (TF-IDF) technique. This process is pivotal as it converts the textual information within the URLs into numerical representations, capturing the essential features necessary for training machine learning models. This vectorization process is applied to both the training and testing sets, ensuring consistency and accuracy in subsequent model evaluations. Moving forward, the research involves the training and evaluation of machine learning models, such as XGBoost and LightGBM classifiers, on the vectorized URLs. These models are trained using the training set and evaluated on the test set, with performance metrics, such as accuracy and a classification report, being generated to assess their efficacy in classifying URLs as phishing or benign. TF-IDF, a staple in natural language processing, works wonders in deciphering term importance within documents. The method's two components, Term Frequency (TF) and Inverse Document Frequency (IDF), team up to gauge a term's significance relative to a document and across an entire document collection. TF, the initial part, gauges how frequently a term pops up within a specific document. It takes into account the ratio of the term's occurrences to the total number of terms in that document. Meanwhile, IDF steps in to spotlight the uniqueness of a term in the entire corpus. It flags rare terms, assigning them higher scores, and thereby emphasizing their distinctiveness across documents. Combining TF and IDF scores through a straightforward multiplication process generates the TF-IDF score, indicating a term's weightage within a document. This score aids in document ranking, relevance assessment in response to search queries, and effective identification of terms specific to a document but not widely spread across the corpus.

Our study focuses on evaluating various machine learning models, notably achieving distinct accuracies: the first classifier used: the XGBoost classifier demonstrates an accuracy of 92.09%, while the LightGBM classifier performs at a higher accuracy of 93.29%. These models are specifically applied in the classification of vectorized URLs. The methodology involves training the models using a dedicated training dataset and subsequently assessing their performance on an independent test dataset. The evaluation encompasses the utilization of key performance metrics, prominently including accuracy, along with the generation of a detailed classification report. The primary aim of this assessment is to gauge the efficacy and precision of these classifiers in accurately categorizing URLs into 'phishing' or 'benign' categories. Simultaneously, a Graph Neural Network (GNN) model is constructed and trained. The graph is formed based on the dataset, with nodes representing URLs and edges formed based on specific conditions related to URL attributes. Features such as URL length, presence of dots, slashes, numbers, special characters, and parameters are meticulously extracted for the GNN model's training. The architecture of the GNN model incorporates graph convolution and fully connected layers, enabling the model to recognize patterns and relationships within the graph structure of URLs. This model is then trained using the constructed graph and the extracted features. The training loop optimizes the model's parameters using the Adam optimizer and minimizes the cross-entropy loss, ensuring the GNN model learns and adapts effectively to classify URLs.

An integral aspect of the research involves real-time monitoring and continuous learning. The system is designed to actively monitor incoming URLs in real-time for potential phishing threats. The machine learning models, including XGBoost, LightGBM, and the GNN, are adaptive and continuously refine their knowledge with the continuous influx of new data. This dynamic approach ensures that the models evolve and adapt to new threats, enhancing their accuracy in identifying potentially malicious URLs. To facilitate user interaction and decision-making, the research integrates a user-friendly interface. This interface provides real-time alerts regarding the classification of URLs, enabling users to take immediate actions when encountering potentially malicious URLs. Moreover, users can input a new URL for prediction. The machine learning models collectively analyze the features of the URL, and the predicted category (benign, phishing, etc.) is displayed, empowering users to make informed decisions about the legitimacy of URLs. Additionally, to ensure the preservation of learned knowledge and to facilitate future use without retraining, the trained GNN model is saved for persistence. This process enables the model to retain its learned knowledge and be loaded and utilized efficiently when necessary.

This work offers users the opportunity to explore and analyze predictions. Users can assess various attributes and characteristics of URLs associated with different threat types. For instance, users can analyze the length of URLs belonging to a specific category in the dataset, providing valuable insights into the characteristics of URLs associated with different threat types, thereby aiding in further analysis and understanding of potential threats. Furthermore, the research allows users to explore categorical information within the dataset. This includes converting the 'type' column into categorical data, which facilitates a clearer understanding of the distribution of phishing and benign labels within the dataset. This exploration provides valuable insights into the prevalence and distribution of potential threats within the dataset. In summary, the "Phishing Website Detection" encompasses a comprehensive workflow, starting from the initial stages of data loading and preprocessing, continuing through model training, real-time monitoring, and user interaction. The research is a multifaceted approach that combines machine learning methodologies with real-time monitoring and user interaction, aiming to effectively identify and categorize potential phishing threats in URLs. Apart from LightGBM and XGBoost, the implementation of another tree-based machine-learning algorithm, CatBoost brought significant value to the research. CatBoost, a gradient boosting framework developed by Yandex, distinguishes itself by specializing in handling categorical features seamlessly, offering robustness, scalability, and superior performance. One of the standout features of CatBoost is its intrinsic capability to handle categorical variables without the need for pre-processing, encoding, or additional handling. It employs a novel approach that avoids the typical requirement of one-hot encoding or label encoding for categorical features. This is particularly advantageous as it reduces the effort and complexity in feature engineering, making it suitable for datasets with a mix of categorical and numerical attributes.

CatBoost's performance was evaluated comprehensively across our dataset within our research. Its comparative analysis with other tree-based models like XGBoost and LightGBM allowed for a deeper understanding of its strengths and weaknesses in various scenarios. By systematically comparing its performance metrics, including accuracy, precision, recall, and F1 score, against other models, we gained insights into its relative performance under diverse conditions. For the CatBoost algorithm, the following hyperparameters were employed:

A. Learning Rate (learning_rate): A value of 0.1 was chosen to control the step size during the optimization process. This parameter influences the convergence speed and prevents overshooting.
B. Number of Iterations (iterations): A total of 1200 iterations were set, determining the number of boosting stages for the ensemble. These iterations contribute to the model's learning and complexity.
C. Tree Depth (depth): A depth of 6 was selected to regulate the depth of individual trees within the boosting framework. This parameter governs the complexity of each tree in the ensemble and affects the model's ability to capture intricate patterns in the data.

These hyperparameters were chosen after a series of iterative experiments and cross-validation procedures, ensuring that the chosen values offered a balance between model complexity, learning capacity, and generalization ability. The aim was to prevent overfitting while maximizing predictive

performance across various datasets and problem domains. Using these hyperparameters, we obtained an accuracy of 92.98%.

Also, we tried to implement the Naive Bayes algorithm into our ensemble of machine learning models to harness its unique strengths. Leveraging the MultinomialNB implementation from the Scikit-learn library, we utilized the algorithm's capability to handle multi-class classification tasks efficiently. The workflow involved several crucial steps. We initially trained the Naive Bayes classifier using the fit method on the training dataset, where the algorithm learned the underlying patterns and relationships between the features and their corresponding labels. The X_train_tfidf and y_train represented the training features and labels, respectively, converted into a term frequency-inverse document frequency (TF-IDF) representation.

Following the training phase, we leveraged the trained model to make predictions on the test dataset using the prediction method, enabling us to assess the model's predictive performance. The resultant y_pred_nb contained the predicted labels for the test data.

Evaluation of the Naive Bayes classifier was performed by calculating its accuracy using the accuracy_score function. This metric quantified the model's overall correctness in predicting the test set labels, providing a general overview of its performance. Additionally, the classification_report function facilitated a comprehensive assessment by presenting a detailed breakdown of precision, recall, F1 score, and other key classification metrics for each class in the dataset. This report allowed for a granular understanding of the model's strengths and weaknesses across different classes, aiding in identifying areas for potential improvement or focus. The incorporation of Naive Bayes into our ensemble allowed for a diverse set of models, enriching the predictive capacity of the final ensemble. Its simplicity, efficiency in handling text-based data, and reasonable performance made it a valuable addition, contributing to a more comprehensive and robust predictive framework for our research. We attained an accuracy of 88.86%, on the given dataset.

V. Results and Discussions

In our comprehensive analysis across a diverse array of machine learning algorithms applied to a shared dataset, we delved into the performances of XGBoost, LightGBM, GNN (Graph Neural Network), CatBoost, and Naive Bayes. This meticulous examination aimed to discern the strengths, weaknesses, and relative efficacy of each algorithm in addressing the classification task at hand. XGBoost and LightGBM, two popular gradient boosting frameworks, exhibited robust performances, showcasing high accuracies and F1 scores. Their iterative boosting methodologies and effective handling of complex relationships within the data enabled them to yield strong predictive capabilities. GNN, leveraging graph-based structures and neural networks, demonstrated its proficiency in capturing intricate patterns inherent in relational data, albeit with a nuanced performance affected by graph complexities.

CatBoost, with its unique handling of categorical features, showcased competitive results, underscoring its efficacy in scenarios with mixed data types. Its intrinsic ability to mitigate overfitting and handle categorical variables without encoding proved advantageous. Conversely, Naive Bayes, known for its simplicity and efficiency, showcased a commendable performance, especially in text-based tasks. Its naive assumption of feature independence, though simplistic, exhibited surprising effectiveness.

Throughout this analysis, key metrics including accuracy, precision, recall, and F1 score were scrutinized to discern the algorithms' predictive prowess. Each algorithm presented distinctive trade-offs between computational efficiency, handling of specific data types, and model complexity, contributing to a nuanced understanding of their suitability for different contexts.

This holistic assessment unveiled the strengths and limitations of each algorithm, informing a deeper understanding of their applicability within specific domains and datasets. The comparative analysis facilitated a nuanced appreciation of the trade-offs involved, aiding in algorithm selection based on the inherent characteristics of the dataset and task requirements. In Google Colab environment, we've set up an interactive cell where users can input URLs to analyze whether a URL is phishing or not. This allows for an on-the-spot assessment of whether a provided URL might belong to a phishing domain or potentially harbor malicious intent. The output in the Colab cell promptly showcases the model's prediction regarding the submitted URL, indicating whether it's identified as safe or potentially

malicious. This immediate feedback offers users insights into the model's interpretation of the URL's nature, aiding in understanding how machine learning algorithms can assist in identifying potential threats in online content. While this setup occurs within the Colab environment and lacks a dedicated graphical user interface, it provides an accessible and interactive means for users to explore the capabilities of machine learning in assessing URL safety and security. The cell-based interaction offers a practical demonstration of how these models can aid in cybersecurity evaluations, fostering awareness and knowledge about online threats

In Figure 3, a comparative examination was conducted across five distinct machine-learning algorithms: XGBoost, LightGBM, GNN, Naive Bayes, and CatBoost. The accuracies achieved by each model were diligently assessed and visualized through a bar plot representation. Each model's accuracy score, expressed as a percentage, was plotted along the y-axis against the respective model names on the x-axis. This visualization, exemplified by a bar plot with distinct colors denoting each model, provided a clear and concise overview of their performance in terms of accuracy. The visual depiction facilitated immediate comparisons, allowing for the identification of models that excelled in predictive accuracy and offering insights into their relative strengths within the context of the dataset.

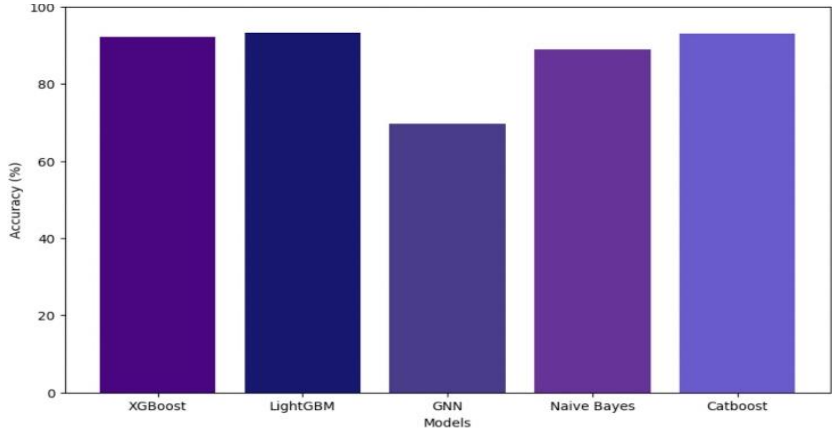

Fig 3. Analysis of model Accuracies

Moreover, a comparative analysis of accuracies was conducted, visualized through a line graph presentation in Figure 4.

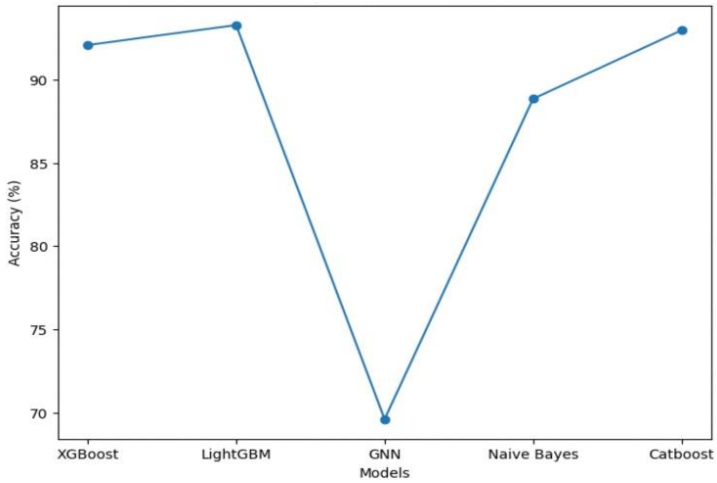

Fig 4. Analysis of model accuracies

Similar to the accuracy analysis, a bar plot was generated to visualize the F1 scores of these models. The F1 scores, representing a harmonic mean of precision and recall, were plotted against the corresponding model names. Distinct colors were utilized to differentiate between models, aiding in easy interpretation and comparison. This graphical representation allowed for a direct assessment of the

models' abilities to balance precision and recall, offering insights into their overall effectiveness in handling both true positives and false negatives.

In Figure 5, the visual portrayal of F1 scores complemented the accuracy analysis, presenting a more holistic view of model performances. It facilitated a comprehensive evaluation, guiding the selection of models best suited for the specific nuances of the dataset and classification task. This comparative analysis of F1 scores empowered informed decision-making regarding the strengths and trade-offs of each model, supporting the identification of models exhibiting a superior balance between precision and recall, crucial for robust and reliable predictions.

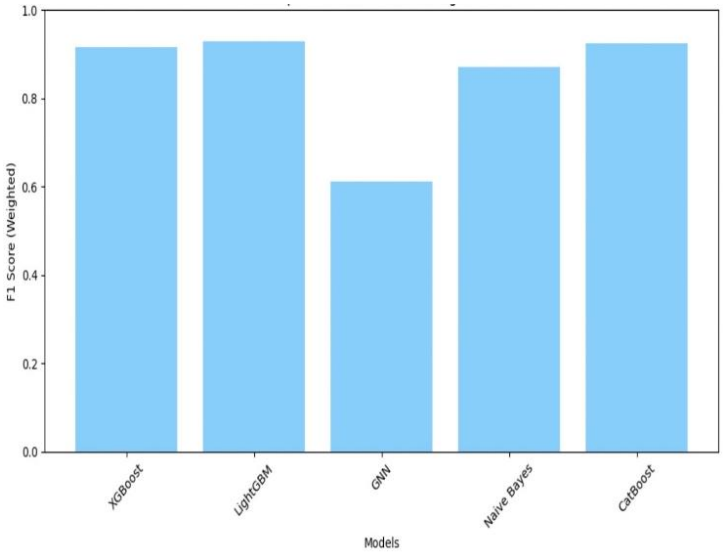

Fig 5. Comparison of F1 Scores among Models

The analysis delved deeper into model evaluation by generating Precision-Recall (PR) curves shown in Figure 6 and 7 respectively for five distinct models—XGBoost, LightGBM, GNN, CatBoost, and Naive Bayes—across different classes within the dataset. These PR curves depicted the trade-off between precision and recall for each model, offering valuable insights into their performance characteristics at varying classification thresholds.

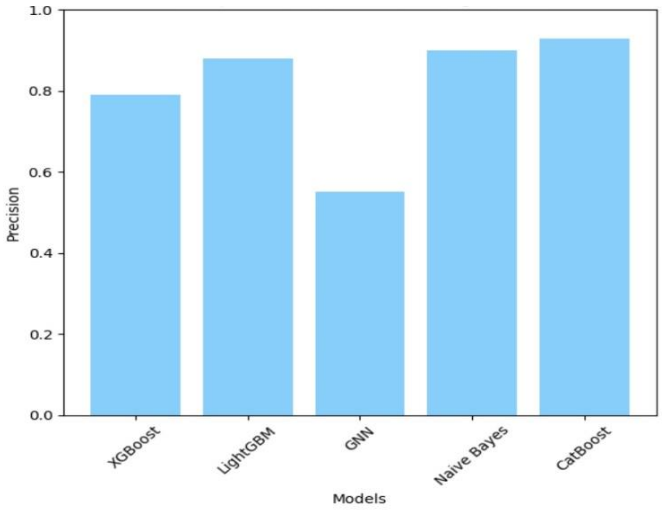

Fig 6. Comparison of Precision among Models

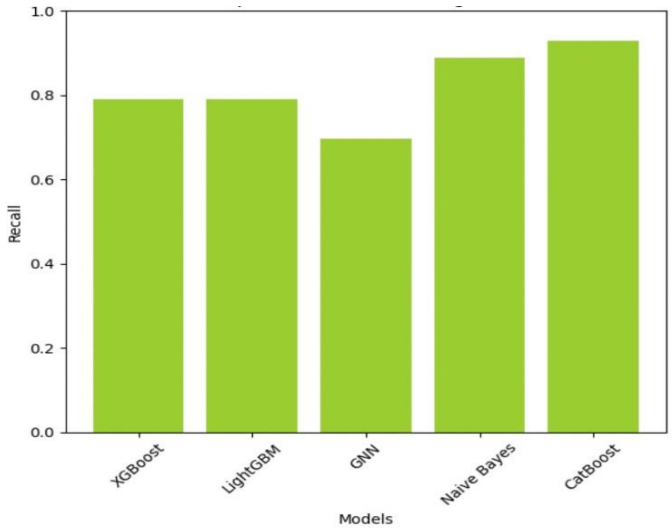

Fig 7. Comparison of Recall among Models

For each class within the dataset, individual PR curves were plotted, illustrating the precision achieved at different levels of recall for every model. The X-axis represented recall, showcasing the model's ability to capture positive instances correctly, while the Y-axis denoted precision, highlighting the proportion of correctly identified positive instances among all instances classified as positive.

In Figure 8, the PR curves provided a comprehensive understanding of each model's ability to maintain high precision while maximizing recall or vice versa. Steeper curves indicated models that achieved higher precision for a given recall threshold, signifying superior performance in classifying positive instances with confidence. On the other hand, flatter curves suggested models that balanced precision and recall more evenly across various thresholds.

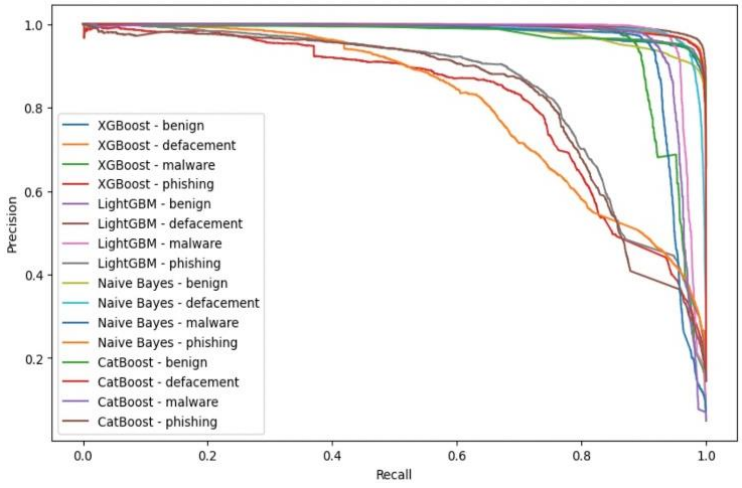

Fig 8. Precision-Recall Curves for Different Models

As per thorough assessment of machine learning models, although GNN's performance left a huge scope for improvement, LightGBM emerged as a standout performer, demonstrating a remarkable precision score of 0.93 alongside a striking recall score of 0.93. This impressive balance between precision and recall highlights LightGBM's ability to accurately classify positive instances while effectively capturing a substantial portion of true positive cases. Following closely behind, XGBoost and CatBoost also exhibited commendable performance, showcasing competitive scores in both precision and recall metrics. Though slightly trailing LightGBM, both XGBoost, and CatBoost displayed strong capabilities in accurately identifying positive instances while maintaining excellent recall levels. Naive Bayes exhibited a slightly lower performance, achieving a precision score of 0.89 and a recall score

of 0.88. This comparative analysis shown in Figure 9 underscores LightGBM's exceptional balance between precision and recall, positioning it as a compelling choice for classification tasks while acknowledging the competitive performances of XGBoost and CatBoost in achieving a harmonious trade-off between precision and recall.

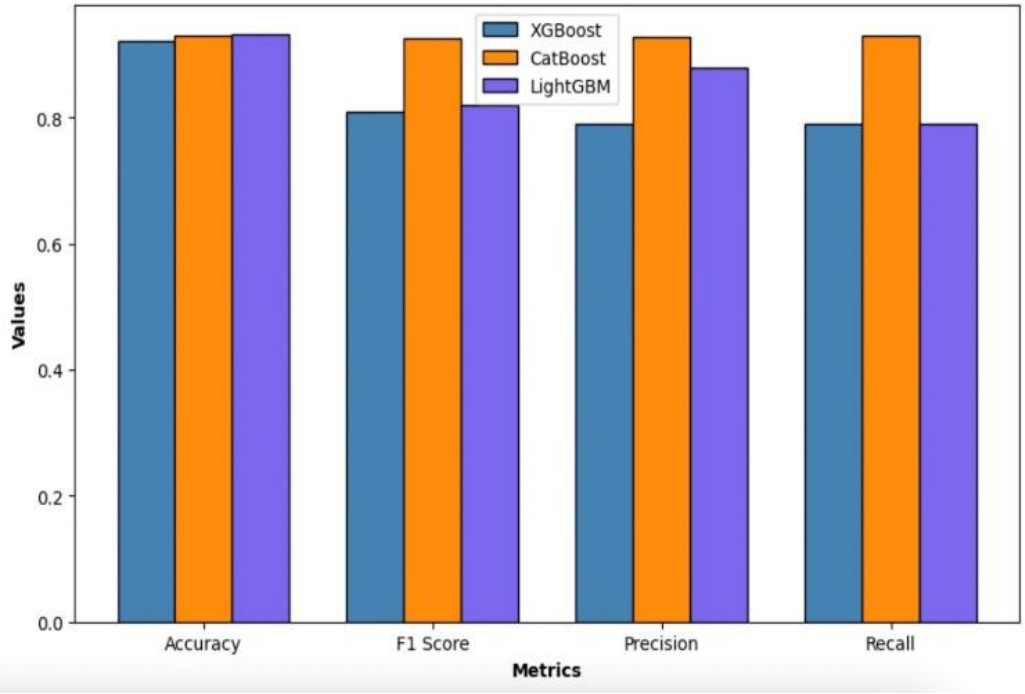

Fig 9. Comparison of Performance Metrics for XGBoost, CatBoost and LightGBM

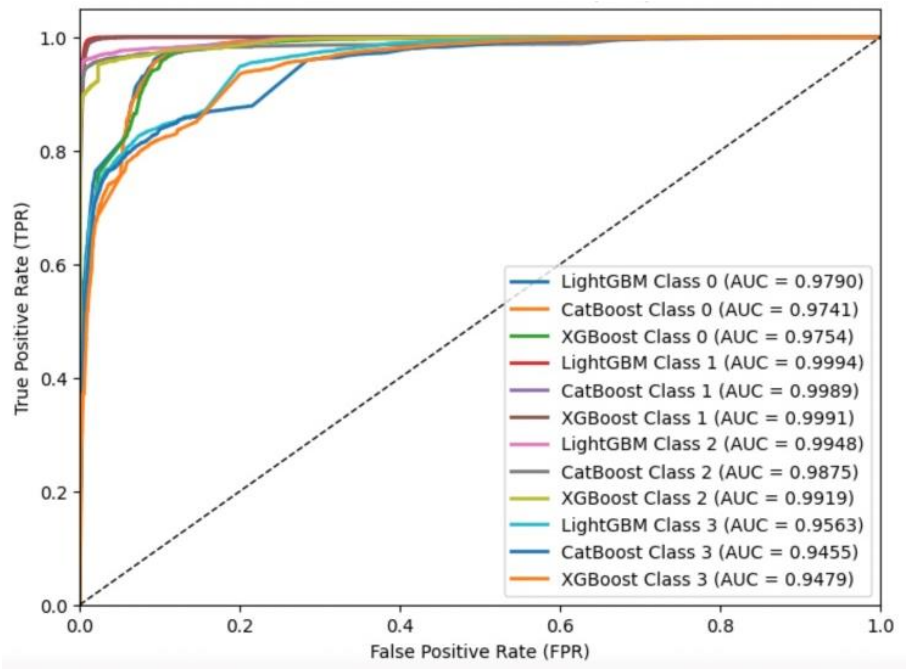

Fig 10. ROC Curve for Each Class (OvR)

This illustration shown in Figure 10 highlights nuanced variations in the Area Under the Curve (AUC) across different models and individual classes. Among these, LightGBM stands out as it demonstrates the most superior performance, showcasing the highest average AUC compared to other models across all classes. Hence, for the Phishing URL classification, all the used classifiers in the experiment perform well for the tree-based models as AUC (area under the ROC curve) is only slightly different for all classifiers, but particularly

Lightgbm is more suitable as it has the highest AUC value in all the used classifiers in our experiment.

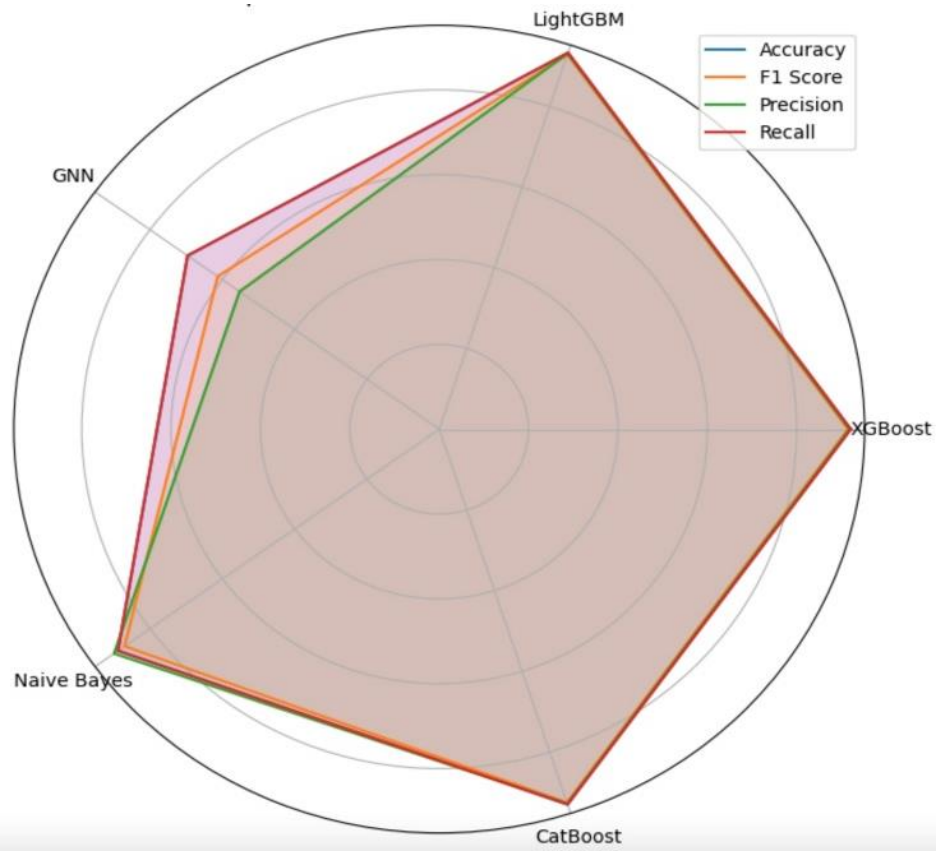

Fig 11. Comparison of Metrics for Different Models

In Figure 11, with respect to the evaluation of various machine learning models applied to our dataset, LightGBM and CatBoost emerged as frontrunners, exhibiting notably high-performance metrics. Both models surpassed the 90% accuracy threshold, signifying their robust predictive capabilities. However, a closer examination revealed nuanced differences in precision and recall metrics, showcasing CatBoost's superiority in these aspects. CatBoost demonstrated higher precision and recall rates compared to LightGBM, contributing to its attainment of the highest F1 score among all models considered. On the other hand, while XGBoost showcased a similar accuracy level to LightGBM, its performance took a hit in precision, experiencing a noticeable decline. Moreover, its recall performance aligned closely with LightGBM. This comparative analysis highlighted the trade-offs among these models, showcasing the unique strengths of CatBoost, the overall balance of LightGBM, and the nuanced shortcomings of XGBoost, enabling a more informed understanding of their performance across varied metrics and emphasizing the importance of considering multiple evaluation criteria in model selection and optimization endeavors. In summary, this work encompasses a comprehensive workflow, starting from the initial stages of data loading and preprocessing, continuing through model training, and precise analysis. The research work is a multifaceted approach combines modern and up-to-date machine learning methodologies with real-time analysis and aims to effectively identify and categorize potential phishing threats in URL.

X. Conclusion And Future Work

In conclusion, "Phishing Website Detection" represents a formidable advancement in the field of phishing website detection, offering users a proactive defense against evolving cyber threats. The research work presented here is a utilization of diverse machine learning models, including XGBoost, LightGBM, Naïve Bayes, CatBoost and a Graph Neural Network (GNN), underscores its commitment to comprehensive and adaptive security measures. By continuously refining its algorithms, embracing user feedback, and staying abreast of emerging technologies, "Phishing Website Detection" positions itself as a resilient and

user-centric solution.

Looking ahead, the scalability and collaborative aspects of the research signal a readiness to tackle future challenges. As the digital landscape continues to evolve, the adaptability of "Phishing Website Detection" ensures that it remains a robust defense mechanism against the sophisticated tactics employed by malicious actors. With a focus on user engagement, technological innovation, and proactive defense strategies, the work stands as a testament to the ongoing commitment to cybersecurity and the protection of users in an increasingly interconnected digital world. The trajectory for advancing this research demonstrates promising avenues for refinement and expansion, marking a critical juncture in its evolution. A central focus involves augmenting the precision of the existing models, notably the Graph Neural Network (GNN), which has showcased the potential for further enhancement. Enhancing the accuracy of these models stands as a paramount objective, with opportunities abounding for refinement and optimization. Envisioning the development of a more intricate model ensemble emerges as a fundamental facet in this pursuit. The amalgamation of disparate models, such as XGBoost, LightGBM, Naïve Bayes, CatBoost and the Graph Neural Network (GNN), within an ensemble learning framework holds significant promise. Leveraging ensemble learning strategies offers a synergistic amalgamation capable of amplifying the overall predictive efficacy. By leveraging the unique strengths and diverse perspectives of these models, a collective predictive framework could achieve heightened accuracy and robustness in detecting phishing websites. Moreover, a forward-looking strategy entails the exploration and integration of state-of-the-art deep learning architectures. This approach stands poised to unlock further insights and advancements in detecting nuanced phishing tactics. Facilitating active user engagement through well-structured feedback mechanisms becomes instrumental in fostering a symbiotic relationship between user input and algorithmic refinement, contributing to a more adaptive and responsive system. Furthermore, ensuring the adaptability of the system to rapidly evolving threats remains a cornerstone. The dynamic adaptation to emerging threats through continuous learning mechanisms will bolster the "Phishing Website Detection" work as a resilient and ever-evolving defense mechanism against the persistent challenges posed by phishing attacks. This holistic approach, incorporating model sophistication, user engagement, and adaptability, promises to fortify the efficacy in safeguarding against the multifaceted landscape of cyber threats.

XI. References

[1]. Basit, A., Zafar, M., Liu, X. et al. A comprehensive survey of AI-enabled phishing attacks detection techniques. Telecommun Syst 76, 139–154 (2021). https://doi.org/10.1007/s11235-020-00733-2

[2]. Suman, Om Prakash, A Novel Approach for Malicious Domain Classification Based on Dns Traffic Analysis and Machine Learning. Available at SSRN: https://ssrn.com/abstract=4592811 or http://dx.doi.org/10.2139/ssrn.4592811

[3]."A.A. Orunsolu, A.S. Sodiya, A.T. Akinwale, A predictive model for phishing detection, Journal of King Saud University - Computer and Information Sciences,Volume 34, Issue 2, 2022, Pages 232-247, ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2019.12.005.

[4]. "Asadullah Safi, Satwinder Singh, A systematic literature review on phishing website detection techniques,Journal of King Saud University - Computer and Information Sciences, Volume 35, Issue 2, 2023, Pages 590-611, ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2023.01.004.(https://www.sciencedirect.com/science/article/pii/S1319157823000034)"

[5]. Dattaa, S., Sena, S. and Kundua, P., A Trustworthy Swift Weapon to Detect the Phishing URLs by Machine Learning Approaches.

[6]. "Dong-Jie Liu, Guang-Gang Geng, Xiao-Bo Jin, Wei Wang, An efficient multistage phishing website detection model based on the CASE feature framework: Aiming at the real web environment, Computers & Security, Volume 110,2021, 102421, ISSN 0167-4048,

https://doi.org/10.1016/j.cose.2021.102421.

[7]. Atlam HF, Oluwatimilehin O. Business Email Compromise Phishing Detection Based on Machine Learning: A Systematic Literature Review. Electronics. 2023; 12(1):42. https://doi.org/10.3390/electronics12010042

[8]. Omari, Kamal. (2023). Comparative Study of Machine Learning Algorithms for Phishing Website Detection. International Journal of Advanced

Computer Science and Applications. 14. 10.14569/IJACSA.2023.0140945.

[9]. "Abdulhamit Subasi, Emir Kremic, Comparison of Adaboost with MultiBoosting for Phishing Website Detection, Procedia Computer Science, Volume 168,2020, Pages 272-278, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.02.251.

[10]. N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma and H. Fujita, "Deep Learning for Phishing Detection: Taxonomy, Current Challenges and Future Directions," in IEEE Access, vol. 10, pp. 36429-36463, 2022, doi: 10.1109/ACCESS.2022.3151903.

[11] Takale, Sayli & Pawar, Samta & Khot, Varad & Acharya, Aditya. (2023). Deep Learning Algorithms for Cybersecurity Applications.

[12]. Alnemari S, Alshammari M. Detecting Phishing Domains Using Machine Learning. Applied Sciences. 2023; 13(8):4649. https://doi.org/10.3390/app13084649

[13]. Deshpande, A., Pedamkar, O., Chaudhary, N. and Borde, S., 2021. Detection of phishing websites using Machine Learning. International Journal of Engineering Research & Technology (IJERT), 10(05).

[14]. A. Aldo Tenis and R. Santhosh, "Modelling an efficient url phishing detection approach based on a dense network model," Computer Systems Science and Engineering, vol. 47, no.2, pp. 2625–2641, 2023.

[15]. "SK Hasane Ahammad, Sunil D. Kale, Gopal D. Upadhye, Sandeep Dwarkanath Pande, E Venkatesh Babu, Amol V. Dhumane, Mr. Dilip Kumar Jang Bahadur, Phishing URL detection using machine learning methods, Advances in Engineering Software, Volume 173, 2022, 103288, ISSN 0965-9978, https://doi.org/10.1016/j.advengsoft.2022.103288.

[16]. Chawla, A. (2022). Phishing website analysis and detection using Machine Learning. International Journal of Intelligent Systems and Applications in Engineering, 10(1), 10–16. https://doi.org/10.18201/ijisae.2022.262

[17]. J. Kumar, A. Santhanavijayan, B. Janet, B. Rajendran and B. S. Bindhumadhava, "Phishing Website Classification and Detection Using Machine Learning," 2020 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2020, pp. 1-6, doi: 10.1109/ICCCI48352.2020.9104161.

[18]. M. Sánchez-Paniagua, E. F. Fernández, E. Alegre, W. Al-Nabki and V. González-Castro, "Phishing URL Detection: A Real-Case Scenario Through Login URLs," in IEEE Access, vol. 10, pp. 42949-42960, 2022, doi: 10.1109/ACCESS.2022.3168681.

[19]. Rendall, K.; Nisioti, A.; Mylonas, A. Towards a Multi-Layered Phishing Detection. Sensors 2020, 20, 4540. https://doi.org/10.3390/s20164540

[20]. G. Sonowal, "A Model for Detecting Sounds-alike Phishing Email Contents for Persons with Visual Impairments," 2020 Sixth International Conference on e-Learning (econf), Sakheer, Bahrain, 2020, pp. 17-21, doi: 10.1109/econf51404.2020.9385451.

[21]. S. Sindhu, S. P. Patil, A. Sreevalsan, F. Rahman and M. S. A. N., "Phishing Detection using Random Forest, SVM and Neural Network with Backpropagation," 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), Bengaluru, India, 2020, pp. 391-394, doi: 10.1109/ICSTCEE49637.2020.9277256.

[22]. C. Singh and Meenu, "Phishing Website Detection Based on Machine Learning: A Survey," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2020, pp. 398-404, doi: 10.1109/ICACCS48705.2020.9074400.

[23]. A. Lakshmanarao, P. S. P. Rao and M. M. B. Krishna, "Phishing website detection using novel machine learning fusion approach," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 2021, pp. 1164-1169, doi: 10.1109/ICAIS50930.2021.9395810.

[24]. R. Zieni, L. Massari and M. C. Calzarossa, "Phishing or Not Phishing? A Survey on the Detection of Phishing Websites," in IEEE Access, vol. 11, pp. 18499-18519, 2023, doi: 10.1109/ACCESS.2023.3247135.

[25]. E. S. Gualberto, R. T. De Sousa, T. P. De Brito Vieira, J. P. C. L. Da Costa and C. G. Duque, "The Answer is in the Text: Multi-Stage Methods for Phishing Detection Based on Feature Engineering," in IEEE Access, vol. 8, pp. 223529-223547, 2020, doi: 10.1109/ACCESS.2020.3043396.

[26] NN Sakhare, SS Imambi, S Kagad, H Malekar, M Dalal, “Stock market prediction using sentiment analysis” International Journal of Advanced Science and Technology, Vol. 4, issue 3, 2020.

[27] NN Sakhare, SA Joshi, “Criminal Identification System Based On Data Mining” 3rd ICRTET, ISBN, Issue 978-93, Pages 5107-220, 2015

[28] NN Sakhare, SA Joshi, “Classification of criminal data using J48-Decision Tree algorithm” IFRSA International Journal of Data Warehousing & Mining, Vol. 4, 2014.

[29] NN Sakhare, SS Imambi, Technical Analysis Based Prediction of Stock Market Trading Strategies Using Deep Learning and Machine Learning Algorithms, International Journal of Intelligent Systems and Applications in Engineering, 2022, 10(3), pp. 411–42.

[30] Sakhare,N.N., Shaik,I.S.,Saha,S.: Prediction of stock market movement via technical analysis of stock data stored on blockchain using novel History Bits based machine learning algorithm. IET Soft.1–12(2023). https://doi.org/10.1049/sfw2.1209212

[31] Sharma, R., Dhabliya, D. A review of automatic irrigation system through IoT (2019) International Journal of Control and Automation, 12 (6 Special Issue), pp. 24-29.

[32] Sharma, R., Dhabliya, D. Attacks on transport layer and multi-layer attacks on manet(2019) International Journal of Control and Automation, 12 (6 Special Issue), pp. 5-11.