ENHANCED PHISHING DETECTION THROUGH HYBRID MACHINE LEARNING AND URL ANALYSIS

[1]K.VISWANATH, Assistant Professor, Dept. of CSE
[2]TALARI SUREKHA, MCA Student
Department of Master of Computer Application
[1, 2] Rajeev Gandhi Memorial College of Engineering and Technology, Nandyal, 518501
Andhra Pradesh, India.

Abstract: Phishing attacks represent a significant and highly dangerous form of cybercrime that occurs on the internet. The project leverages a dataset containing phishing URLs. These URLs are the web addresses used by cybercriminals to carry out phishing attacks. To detect and combat these phishing attempts effectively, the project employs a variety of machine learning algorithms. These include decision tree, linear regression, random forest, naive Bayes, gradient boosting classifier, K-neighbors classifier, support vector classifier, and a hybrid model referred to as LSD. In addition to using these algorithms, the project incorporates advanced techniques such as cross-fold validation and Grid Search Hyperparameter Optimization. To determine how well these models work, the project uses specific evaluation metrics. These metrics include precision, accuracy, recall, and F1-score. In enhancing the Phishing Detection System, a Stacking Classifier with RF + MLP using LightGBM exhibits improved performance.

Index terms - *Voting classifier, ensemble classifier, machine learning, uniform resource locator (URL), logistic regression, support vector machine, and decision tree (LSD), protocol, cyber security, social networks.*

1. INTRODUCTION

The internet plays a crucial role in various aspects of human life. The Internet is a collection of computers connected through telecommunication links such as phone lines, fiber optic lines, and wireless and satellite connections. It is a global computer network. The internet is used to obtain information stored on computers, which are known as hosts and servers. For communication purposes, they used a protocol called Internet protocol/transmission control protocol (IP-TCP). The government is not recognized as an owner of the Internet; many organizations, research agencies, and universities participate in managing the Internet. This has led to many convenient experiences in our lives regarding entertainment, education, banking, industry, online freelancing, social media, medicine, and many other fields in daily life. The internet provides many advantages in different fields of life.

In the field of information search, the Internet has become a perfect opportunity to search for data for educational and research purposes. Email is a messaging source in fast way on the Internet through which we can send files, videos, pictures, and any applications, or write a letter to another person around the world. E-commerce is also used on the internet. People can conduct business and financial deals with customers worldwide through e-commerce. Online results are helpful in displaying results online and have become a more useful source of the covid-19 pandemic in 2020. Many classes and business meetings are performed online, which requires time and is fulfilled through the internet. Owing to the increase in data sharing, the chances of loss and cyber-attack also increase.

Online shopping is the biggest Internet use that helps traders sell projects online worldwide. Amazon operates a large online sales system. Fast communication is performed through the Internet, which is currently used through Facebook, Instagram, WhatsApp, and other social networks, making communication fast and easily available. Therefore, it is necessary to maintain a privacy policy in which communication and its users cannot be defective. The Internet provides a great opportunity for attackers to engage in criminal activities such as online fraud, malicious software, computer viruses, ransomware, worms, intellectual property rights, denial of service attacks, money laundering, vandalism, electronic terrorism, and extortion.

Hacking is a major destroyer of the Internet through which any person can hack computer information and use it in different ways to harm others. Immorality, which harms moral values, is a major issue for the younger generations. Detecting these websites rather than websites that appear simple and secure, will help people. Therefore, an awareness of these websites is necessary. Viruses can damage an entire computer network and confidential information by spreading to multiple computers. It is not suitable to use unauthorized websites on the internet. Phishing detection is required for all of these aspects to secure our computer system. Cyber security has become a major global issue.

Over the last decade, several anti-phishing detection mechanisms have been proposed. These studies have mainly focused on the structure of a uniform resource locator (URL) based on feature-selection methods for machine learning. Berners-Lee (1994) developed the URL. The format of the URL is defined by preexisting sources and protocols. Pre-existing systems, such as domain names with syntax of file paths, were created and proposed in 1985. Slashes were used to separate the filenames and directories from the path of a file. Double slashes were used to separate the server names and file paths. Berners-Lee then introduced dots to separate the domain names. HTTP URL consists of a syntax which is divided into five components which are in hierarchical sequence.

2. LITERATURE SURVEY

A traditional settlement is defined by a colony in which physical attributes and its occupants retain their daily traditions and skills, as well as other cultural practices. However, due to urbanization and economic development, certain traditional settlements in Malaysia are currently undergoing tremendous changes. Hence, this paper identifies the physical attributes that are significant for the preservation of social sustainability in the traditional Malay settlement. In this research, a qualitative methodology was used to identify the characteristics of the traditional settlements in Kuala Terengganu. In this study [1], street pattern, housing boundaries and open spaces were identified to be significant as key characteristics for the preservation of social interaction in the three traditional settlements studied. Therefore, the study concluded that methods and selections of such physical characteristics and space typology are significant in order to maintain the social sustainability in traditional settlement communities.

To address the evolving strategies and techniques employed by hackers, intrusion detection systems (IDS) is required to be applied across the network to detect and prevent against attacks. Appropriately, each [3, 4] TCP/IP network layers has specific type of network attacks that means each network layer needs a specific type of IDS. Now-a -days Machine Learning becomes most powerful tool to deal with network security challenges given that the network level data generated is huge in volume and decision related to attacks need to be decided with high speed and accuracy. Classification is one of the techniques to deal with new and unknown attacks with network intrusion using machine learning. In this chapter [4], we detect the normal and anomaly attacks of the TCP/IP [3] packets from publicly available training dataset using Gaussian Naive Bayes, logistic regression, Decision Tree and artificial neural network on intrusion detection systems. Using CoLab environment, we provide some experimental results showing that Decision tree performed better than Gaussian Naïve Bayes, Logistic regression and Neural Network with a publicly available dataset.

Understanding the evolving characteristics of the World Wide Web is challenging due to its immense size and diversity. In this paper [5], we investigate Web structure and dynamics by analyzing over 1 trillion URLs requested during Web browsing by a 2 million person user panel over a period of 12 months. We begin by examining the lifetime of URLs [5, 6, 13, 20] and find that in contrast to early studies, the set of URLs visited is highly dynamic and well-modeled by a gamma distribution. Next, we analyze URL-traversal patterns and find that browsing behaviors differ substantially from hyperlink connectivity. One consequence of this is that the structure of the Web that is derived from hyperlink connectivity does not extend directly to actual user behavior. Finally, we consider the commonly used path and query portions of URLs [32] and highlight their characteristics when used by different website genres. These semantic differences suggest that URL structure can broadly classify the kind of resource that a URL references [34, 36]. Our analyses lead to a set of proposed enhancements to the URL standard that would improve Web manageability and transparency and make a step toward the semantic web.

Since decades, several attempts have been made on Web based research particularly based on HTML web pages because of their more availability. But, W3 consortium stated that HTML [29] do not provide a better description of semantic structure of the web page contents, because of its limited pre-defined tags, semi structured data, case sensitivity and so on. To overcome these draw backs, Web developers started to develop Web page(s) on XML, Flash kind of new technologies. It makes a way for new research methods. In this article [6], we mainly focuses on XML URL classification based on their semantic structure orientation. Experimental results show that proposed method achieves overall accuracy level of 97.36% of classification.

This research focuses on evaluating whether a website is legitimate or phishing [7]. Our research contributes to improving the accuracy of phishing website detection. Hence, a feature selection algorithm is employed and integrated with an ensemble learning methodology, which is based on majority voting, and compared with different classification models including Random forest, Logistic Regression, Prediction model etc. Our research demonstrates that current phishing detection technologies have an accuracy rate between 70% and 92.52%. The experimental results prove that the accuracy rate of our proposed model can yield up to 95%, which is higher than the current technologies for phishing website detection. Moreover, the learning models used during the experiment indicate that our proposed model has a promising accuracy rate.

With the advent of online social media, phishers have started using social networks like Twitter, Facebook, and Foursquare to spread phishing scams. Twitter is an immensely popular micro-blogging network where people post short messages of 140 characters called tweets. It has over 100 million active users who post about 200 million tweets everyday. Phishers have started using Twitter as a medium to spread phishing because of this vast information dissemination. Further, it is difficult to detect phishing on Twitter unlike emails because of the quick spread of phishing links in the network, short size of the content, and use of URL obfuscation to shorten the URL [15, 16, 17]. Our technique, PhishAri, detects phishing on Twitter in realtime. We use Twitter specific features along with URL features to detect whether a tweet posted with a URL is phishing or not. Some of the Twitter specific features we use are tweet content and its characteristics like length, hashtags, and mentions. Other Twitter features used are the characteristics of the Twitter user posting the tweet such as age of the account, number of tweets, and the follower-followee ratio. These twitter specific features coupled with URL based features prove to be a strong mechanism to detect phishing tweets. [8] We use machine learning classification techniques and detect phishing tweets with an accuracy of 92.52%. We have deployed our system for end-users by providing an easy to use Chrome browser extension. The extension works in realtime and classifies a tweet as phishing or safe. In this research, we show that we are able to detect phishing tweets at zero hour with high accuracy which is much faster than public blacklists and as well as Twitter's own defense mechanism to detect malicious content. [8] We also performed a quick user evaluation of PhishAri in a laboratory study to evaluate the usability and effectiveness of PhishAri and showed that users like and find it convenient to use PhishAri in real-world. To the best of our knowledge, this is the first realtime, comprehensive and usable system to detect phishing on Twitter.

3. METHODOLOGY

i) Proposed Work:

The proposed system uses a hybrid machine learning approach to detect phishing attacks through URL attributes. It leverages various machine learning algorithms, including decision trees, random forests, and more, to enhance accuracy. Techniques like cross-fold validation and hyperparameter optimization further improve its effectiveness in distinguishing phishing [15] URLs, ensuring robust protection against cyber threats. And also added to the Phishing Detection System project, a Stacking Classifier integrating Random Forest (RF) and Multilayer Perceptron (MLP) with LightGBM has been employed, demonstrating improved performance in phishing detection. Additionally, a user-friendly Flask framework with SQLite integration has been developed, incorporating secure signup and signin functionalities for effective user testing and enhancing the practical usability of the system in real-world scenarios.

ii) System Architecture:

- The system starts with a dataset containing URLs [15, 16, 17], labeled as either phishing or legitimate. This dataset serves as the foundation for training and testing the machine learning models.
- Before processing the data, any null or missing values in the dataset are removed or appropriately handled to ensure data quality and consistency.
- Feature engineering is performed to convert the URLs into numerical feature vectors. Various attributes of the URLs, such as domain length, presence of special characters, and more, are transformed into numerical representations. These feature vectors are used as input data for the machine learning models.
- The dataset is split into two parts: training data and testing data. The training data is used to train the machine learning models, while the testing data is reserved for evaluating model performance.
- Several machine learning algorithms are employed to build predictive models for phishing detection. These models include decision trees, random forests, support vector machines, and more. We have also built stacking classifier as an extension to the project. Each model learns to recognize patterns and characteristics associated with phishing URLs [26].
- After training, the machine learning models become capable of classifying URLs as either phishing or legitimate based on the feature vectors. These trained models are saved for future use and integration into the detection system.
- The system uses evaluation metrics to assess the performance of the trained models on the testing data. These metrics help gauge how well the models can identify phishing URLs [17] while minimizing false positives and false negatives.

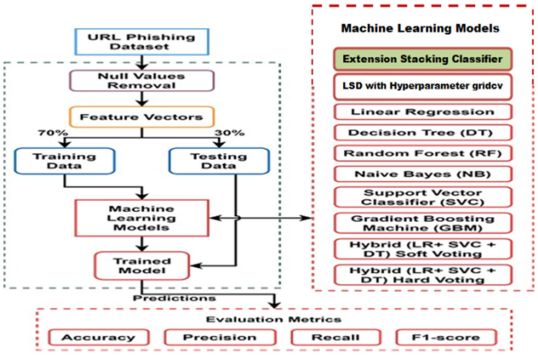
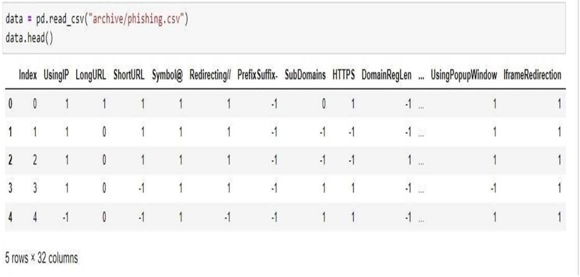


Fig 1 Proposed architecture

iii) Dataset collection:

We employ exploratory data analysis, and feature correlation analysis to better understand the Phishing URL Feature Data [27]. These techniques help reveal data distributions, outliers, and relationships between variables, aiding in subsequent data processing and model building. The dataset used in the proposed system is called the "URL-based phishing dataset" and was extracted from a well-known dataset repository called Kaggle. It consists of phishing and legitimate URLs [15, 16, 17] collected from over 11,000 websites, presented in vector form.



```
data = pd.read_csv("archive/phishing.csv")
data.head()
```

```
   Index  UsingIP  LongURL  ShortURL  Symbol@  Redirecting//  PrefixSuffix-  SubDomains  HTTPS  DomainRegLen  ...  UsingPopupWindow  IframeRedirection
0      0        1        1         1        1              1             -1           0      1            -1  ...                 1                  1
1      1        1        0         1        1              1             -1          -1     -1            -1  ...                 1                  1
2      2        1        0         1        1              1             -1          -1     -1             1  ...                 1                  1
3      3        1        0        -1        1              1             -1           1      1            -1  ...                -1                  1
4      4       -1        0        -1        1             -1             -1           1      1            -1  ...                 1                  1

5 rows × 32 columns
```

Fig 2 Dataset

So, these are the top 5 rows of the phishing data set on which we will train the models, we have extracted it from kaggle. So, it contains 32 columns, we are displaying few of them here.

iv) Data Processing:

Data processing involves transforming raw data into valuable information for businesses. Generally, data scientists process data, which includes collecting, organizing, cleaning, verifying, analyzing, and converting it into readable formats such as graphs or documents. Data processing can be done using three methods i.e., manual, mechanical, and electronic. The aim is to increase the value of information and facilitate decision-making. This enables businesses to improve their operations and make timely strategic decisions. Automated data processing solutions, such as computer software programming, play a significant role in this. It can help turn large amounts of data, including big data, into meaningful insights for quality management and decision-making.

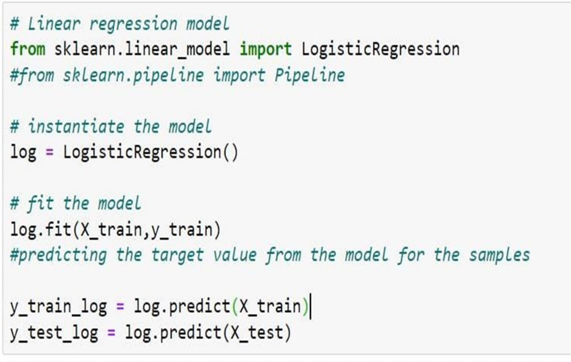
v) Feature selection:

Feature selection is the process of isolating the most consistent, non-redundant, and relevant features to use in model construction. Methodically reducing the size of datasets is important as the size and variety of datasets continue to grow. The main goal of feature selection is to improve the performance of a predictive model and reduce the computational cost of modeling.

Feature selection [8, 9, 10], one of the main components of feature engineering, is the process of selecting the most important features to input in machine learning algorithms. Feature selection techniques are employed to reduce the number of input variables by eliminating redundant or irrelevant features and narrowing down the set of features to those most relevant to the machine learning model. The main benefits of performing feature selection in advance, rather than letting the machine learning model figure out which features are most important.

vi) Algorithms:

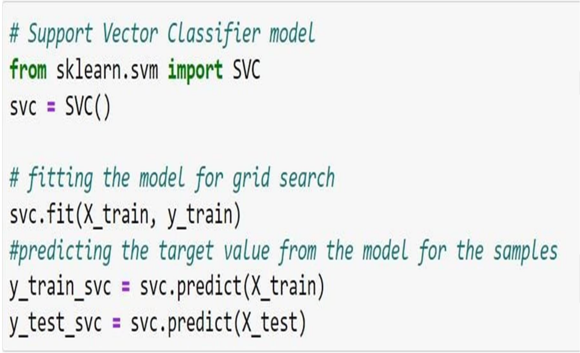
Logistic Regression is a classification algorithm that predicts the probability of an input belonging to a specific category. It employs the sigmoid function to map the input features to a probability score between 0 and 1, and a threshold is applied to classify the input into one of two or more categories based on this probability. The model learns coefficients during training to best fit the data and make accurate classifications.



```
# Linear regression model
from sklearn.linear_model import LogisticRegression
#from sklearn.pipeline import Pipeline

# instantiate the model
log = LogisticRegression()

# fit the model
log.fit(X_train,y_train)
#predicting the target value from the model for the samples

y_train_log = log.predict(X_train)
y_test_log = log.predict(X_test)
```

Fig 3 Linear regression

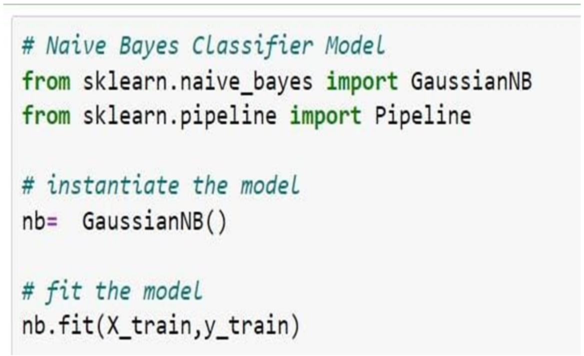
A Support Vector Classifier (SVC) is a machine learning model that finds the best possible boundary (hyperplane) to separate different classes of data while maximizing the margin between them. It identifies key support vectors to make accurate classifications, making it effective for both binary and multi-class classification tasks.



```
# Support Vector Classifier model
from sklearn.svm import SVC
svc = SVC()

# fitting the model for grid search
svc.fit(X_train, y_train)
#predicting the target value from the model for the samples
y_train_svc = svc.predict(X_train)
y_test_svc = svc.predict(X_test)
```

Fig 4 SVC

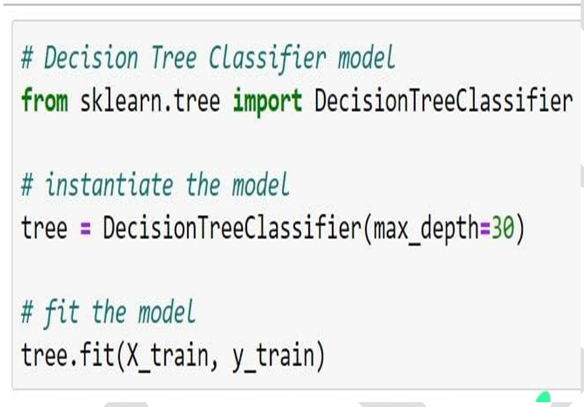
Naive Bayes is a probabilistic classification algorithm that works by applying Bayes' theorem with the "naive" assumption of feature independence. It calculates the probability of a data point belonging to a particular class based on the probabilities of its individual features. Naive Bayes is particularly efficient for text classification tasks, spam detection, and other situations where feature independence is a reasonable approximation [9].



```
# Naive Bayes Classifier Model
from sklearn.naive_bayes import GaussianNB
from sklearn.pipeline import Pipeline

# instantiate the model
nb=  GaussianNB()

# fit the model
nb.fit(X_train,y_train)
```

Fig 5 Naïve bayes

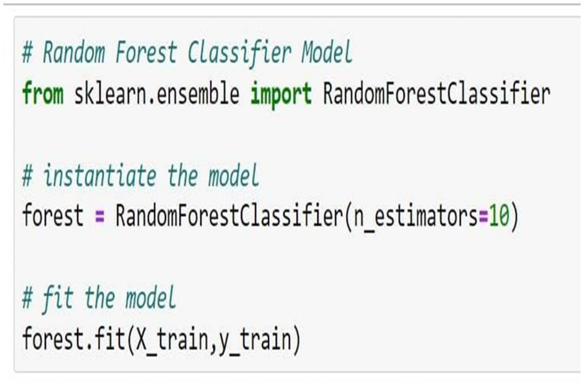
A Decision Tree is a machine learning model that makes decisions by recursively splitting data into subsets based on the most significant feature, aiming to classify or predict outcomes. It creates a tree-like structure where each node represents a feature and each branch represents a possible decision, making it interpretable and effective for various tasks.



```
# Decision Tree Classifier model
from sklearn.tree import DecisionTreeClassifier

# instantiate the model
tree = DecisionTreeClassifier(max_depth=30)

# fit the model
tree.fit(X_train, y_train)
```

Fig 6 Decision tree

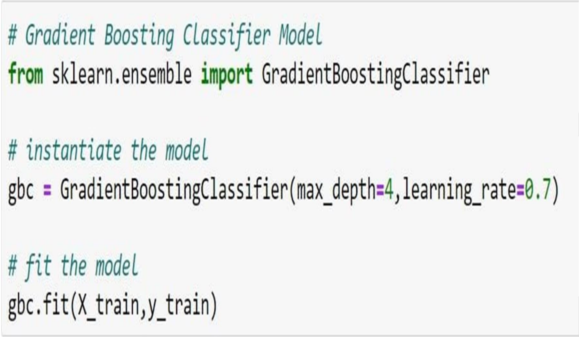
Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. It works by training a collection of decision trees on random subsets of the data and then averaging their predictions. This ensemble approach enhances accuracy, reduces overfitting, and provides robust performance for both classification and regression tasks.



```
# Random Forest Classifier Model
from sklearn.ensemble import RandomForestClassifier

# instantiate the model
forest = RandomForestClassifier(n_estimators=10)

# fit the model
forest.fit(X_train,y_train)
```

Fig 7 Random forest

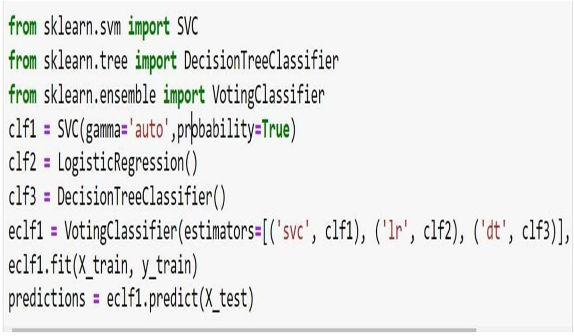
Gradient Boosting is an ensemble machine learning technique that sequentially builds a predictive model by combining the strengths of multiple weak learners, typically decision trees. It does so by focusing on the errors made by the previous models and adjusting its predictions to reduce those errors, ultimately creating a powerful and accurate predictive model that excels in various tasks, including regression and classification.



```
# Gradient Boosting Classifier Model
from sklearn.ensemble import GradientBoostingClassifier

# instantiate the model
gbc = GradientBoostingClassifier(max_depth=4,learning_rate=0.7)

# fit the model
gbc.fit(X_train,y_train)
```

Fig 8 Gradient boosting

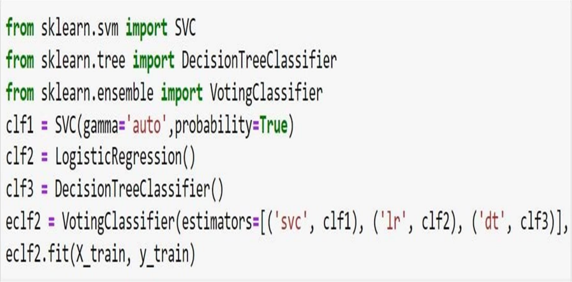
The Hybrid LSD (Soft) model combines Logistic Regression, Support Vector Machine, and Decision Tree using soft voting to classify data. It leverages the strengths of each model to make predictions, with the flexibility to handle different types of data and improve accuracy in classification tasks.



```
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import VotingClassifier
clf1 = SVC(gamma='auto',probability=True)
clf2 = LogisticRegression()
clf3 = DecisionTreeClassifier()
eclf1 = VotingClassifier(estimators=[('svc', clf1), ('lr', clf2), ('dt', clf3)],
eclf1.fit(X_train, y_train)
predictions = eclf1.predict(X_test)
```

Fig 9 Hybrid LSD (soft

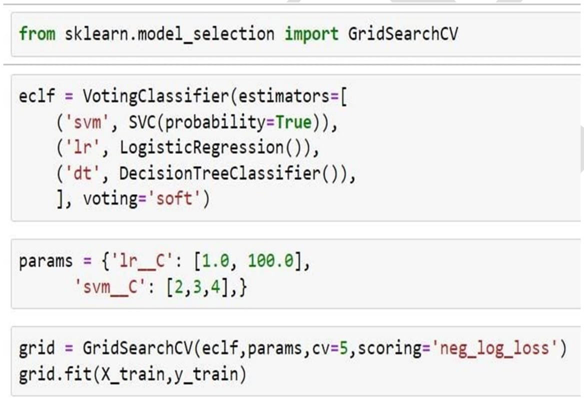
The Hybrid LSD (Hard) model combines Logistic Regression, Support Vector Machine, and Decision Tree algorithms with a hard voting technique to make classification decisions. Each component model contributes its prediction, and the final decision is made by majority voting, enhancing accuracy and robustness in various classification tasks.



```
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import VotingClassifier
clf1 = SVC(gamma='auto',probability=True)
clf2 = LogisticRegression()
clf3 = DecisionTreeClassifier()
eclf2 = VotingClassifier(estimators=[('svc', clf1), ('lr', clf2), ('dt', clf3)],
eclf2.fit(X_train, y_train)
```

Fig 10 Hybrid LSD (Hard)

The LSD (Logistic Regression, Support Vector Machine, Decision Tree) model with Hyperparameter GridCV is a hybrid classification model that combines the strengths of Logistic Regression, Support Vector Machine, and Decision Tree algorithms, enhancing accuracy and efficiency. GridCV systematically searches through hyperparameter combinations to optimize model performance, making it effective in various classification tasks.



```
from sklearn.model_selection import GridSearchCV
```

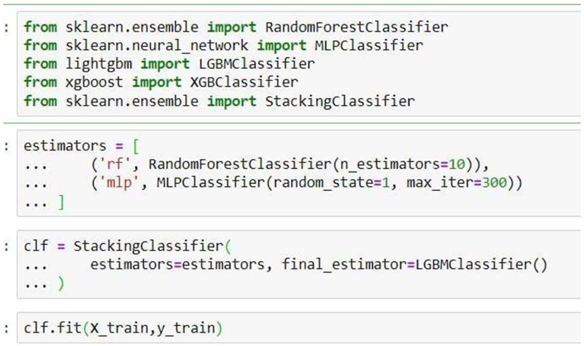
```
eclf = VotingClassifier(estimators=[
    ('svm', SVC(probability=True)),
    ('lr', LogisticRegression()),
    ('dt', DecisionTreeClassifier()),
    ], voting='soft')
```

```
params = {'lr__C': [1.0, 100.0],
      'svm__C': [2,3,4],}
```

```
grid = GridSearchCV(eclf,params,cv=5,scoring='neg_log_loss')
grid.fit(X_train,y_train)
```

Fig 11 LSD with Hyperparameter GridCV

The project employs a StackingClassifier, an ensemble technique, to combine predictions from RandomForestClassifier and MLPClassifier as base classifiers. It uses LGBMClassifier as a meta-estimator to make the final prediction, extending the project's capabilities for improved classification performance.



```
: from sklearn.ensemble import RandomForestClassifier
  from sklearn.neural_network import MLPClassifier
  from lightgbm import LGBMClassifier
  from xgboost import XGBClassifier
  from sklearn.ensemble import StackingClassifier

: estimators = [
  ...     ('rf', RandomForestClassifier(n_estimators=10)),
  ...     ('mlp', MLPClassifier(random_state=1, max_iter=300))
  ... ]

: clf = StackingClassifier(
  ...     estimators=estimators, final_estimator=LGBMClassifier()
  ... )

: clf.fit(X_train,y_train)
```

Fig 12 Stacking classifier

4. EXPERIMENTAL RESULTS

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

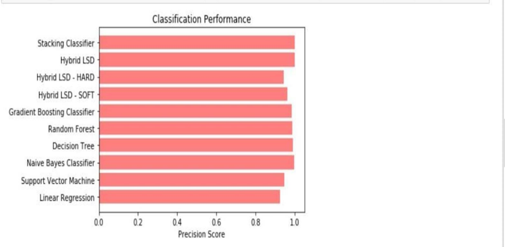


Fig 13 Precision comparison graph

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

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

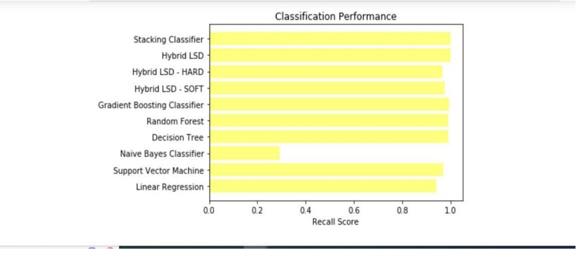


Fig 14 Recall comparison graph

Accuracy: Accuracy is the proportion of correct predictions in a classification task, measuring the overall correctness of a model's predictions.

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

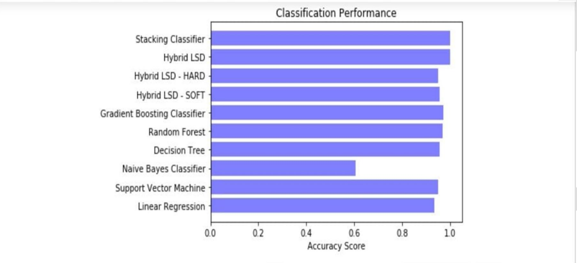


Fig 15 Accuracy graph

F1 Score: The F1 Score is the harmonic mean of precision and recall, offering a balanced measure that considers both false positives and false negatives, making it suitable for imbalanced datasets.

$$\text{F1 Score} = 2 * \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} * 100$$

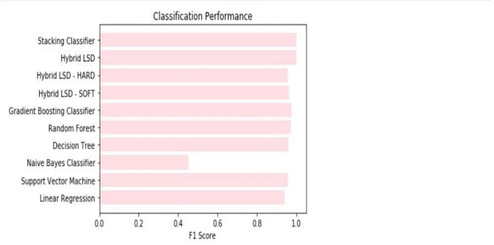
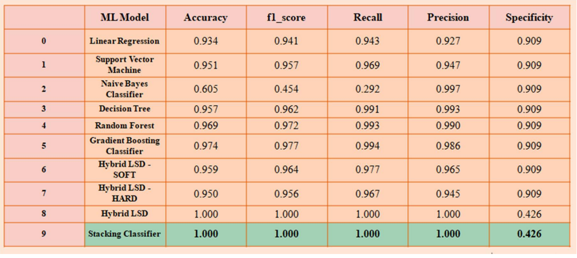
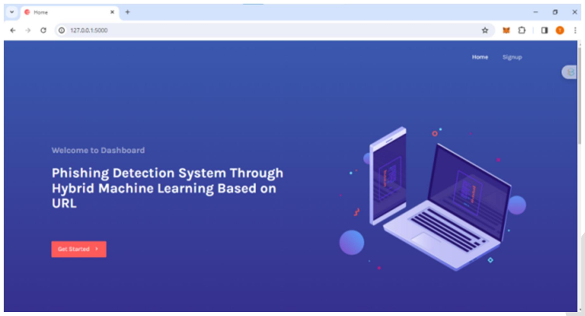


Fig 16 F1Score



	ML Model	Accuracy	f1_score	Recall	Precision	Specificity
0	Linear Regression	0.934	0.941	0.943	0.927	0.909
1	Support Vector Machine	0.951	0.957	0.969	0.947	0.909
2	Naive Bayes Classifier	0.605	0.454	0.292	0.997	0.909
3	Decision Tree	0.957	0.962	0.991	0.993	0.909
4	Random Forest	0.969	0.972	0.993	0.990	0.909
5	Gradient Boosting Classifier	0.974	0.977	0.994	0.986	0.909
6	Hybrid LSD - SOFT	0.959	0.964	0.977	0.965	0.909
7	Hybrid LSD - HARD	0.950	0.956	0.967	0.945	0.909
8	Hybrid LSD	1.000	1.000	1.000	1.000	0.426
9	Stacking Classifier	**1.000**	**1.000**	**1.000**	**1.000**	**0.426**

Fig 17 Performance Evaluation



Welcome to Dashboard

Phishing Detection System Through Hybrid Machine Learning Based on URL

Fig 18 Home page

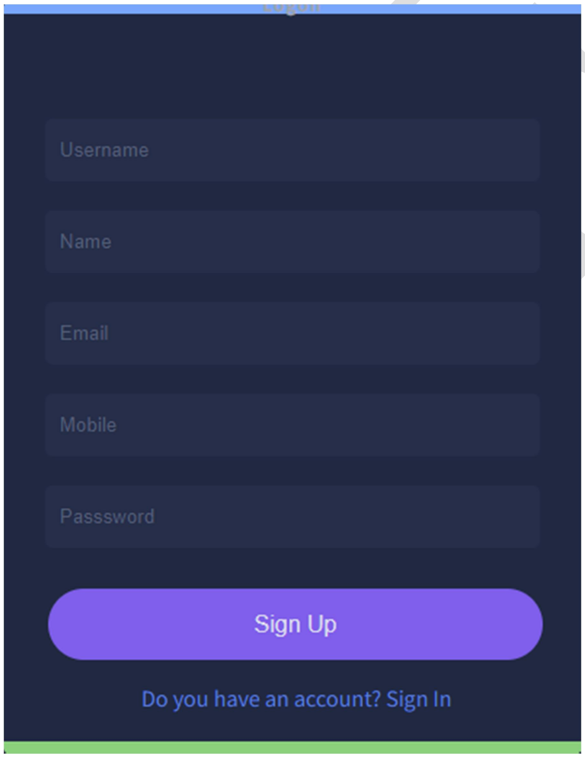
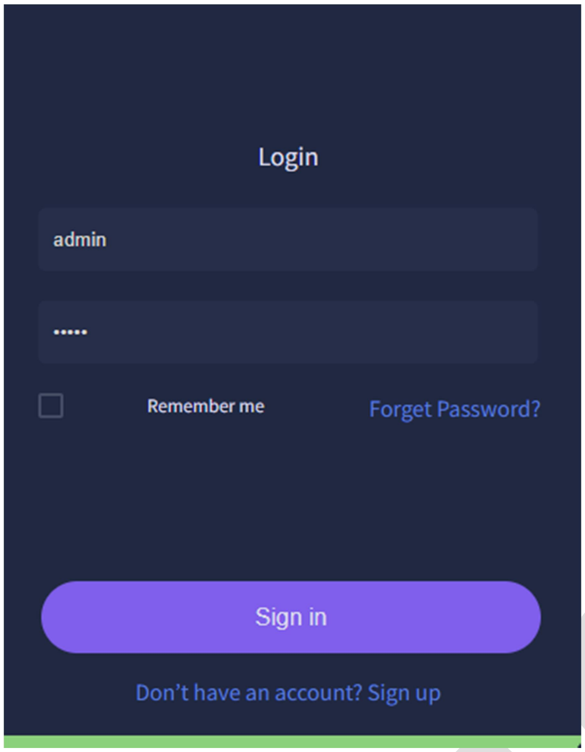


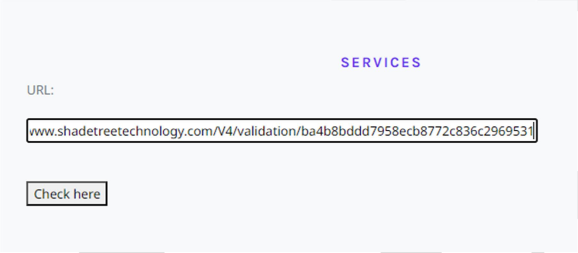
Fig 19 Signin page



admin

•••••

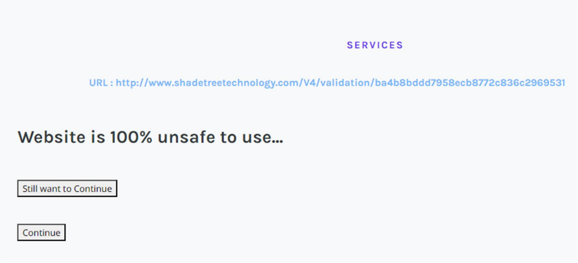
Fig 20 Login page



URL:

ww.shadetreetechnology.com/V4/validation/ba4b8bddd7958ecb8772c836c2969531

Fig 21 User input



URL : http://www.shadetreetechnology.com/V4/validation/ba4b8bddd7958ecb8772c836c2969531

Website is 100% unsafe to use...

Fig 22 Predict result for given input

5. CONCLUSION

The project embraced a hybrid machine learning approach, giving importance to URL attributes to strengthen phishing detection capabilities. Utilizing an array of models including decision tree, random forest, support vector classifier, LSD (both hard and soft), alongside the extension of the stacking classifier method and Hybrid LSD, each showcasing perfect accuracy and F1-score, aimed to improve the system's accuracy and efficiency in identifying phishing threats [2, 3]. And also introduced a stacking classifier, selected for its remarkable accuracy and F-score, marking a substantial improvement in the system's overall effectiveness. This choice reflects a commitment to deploying a high-performing model for reliable phishing detection. The integration of Flask with SQLite ensures a smooth and secure front-end for user testing, enhancing the system's practical usability. This combination facilitates user interactions, offering seamless signup, signin, and testing functionalities. In conclusion, the project's holistic phishing detection system addresses a crucial cybersecurity challenge, providing robust protection against severe phishing attacks [7, 8]. The incorporation of advanced machine learning techniques, model diversity, and user-friendly features highlights the project's dedication to creating an effective solution in the dynamic landscape of cybersecurity threats.

6. FUTURE SCOPE

Future enhancements can involve the exploration of additional algorithms and techniques to further enhance the accuracy and efficiency of phishing detection [11, 12, 13, 16]. By continuously investigating and adopting innovative approaches, the system can stay ahead of emerging threats. Expanding the project to include real-time monitoring capabilities is essential in the face of ever-evolving cyber threats. This proactive approach allows the system to adapt quickly to new attack methods and provide timely protection. To ensure the robustness and generalizability of the model, it can be evaluated on a larger and more diverse dataset. This expanded dataset provides a better representation of real-world scenarios, enhancing the system's effectiveness. Exploring the integration of user behavior and network traffic analysis can significantly enhance the system's detection capabilities. By considering additional data sources, the system can provide more comprehensive and accurate threat assessments. The model's capabilities can extend beyond phishing detection to encompass other cybercrimes such as malware attacks and social engineering techniques. This broader scope enables the system to address a wider range of cybersecurity threats effectively.

REFERENCES

[1] N. Z. Harun, N. Jaffar, and P. S. J. Kassim, ‘‘Physical attributes significant in preserving the social sustainability of the traditional malay settlement,’’ in Reframing the Vernacular: Politics, Semiotics, and Representation. Springer, 2020, pp. 225–238.

[2] D. M. Divakaran and A. Oest, ‘‘Phishing detection leveraging machine learning and deep learning: A review,’’ 2022, arXiv:2205.07411.

[3] A. Akanchha, ‘‘Exploring a robust machine learning classifier for detecting phishing domains using SSL certificates,’’ Fac. Comput. Sci., Dalhousie Univ., Halifax, NS, Canada, Tech. Rep. 10222/78875, 2020.

[4] H. Shahriar and S. Nimmagadda, ‘‘Network intrusion detection for TCP/IP packets with machine learning techniques,’’ in Machine Intelligence and Big Data Analytics for Cybersecurity Applications. Cham, Switzerland: Springer, 2020, pp. 231–247.

[5] J. Kline, E. Oakes, and P. Barford, ‘‘A URL-based analysis of WWW structure and dynamics,’’ in Proc. Netw. Traffic Meas. Anal. Conf. (TMA), Jun. 2019, p. 800.

[6] A. K. Murthy and Suresha, ‘‘XML URL classification based on their semantic structure orientation for web mining applications,’’ Proc. Comput. Sci., vol. 46, pp. 143–150, Jan. 2015.

[7] A. A. Ubing, S. Kamilia, A. Abdullah, N. Jhanjhi, and M. Supramaniam, ‘‘Phishing website detection: An improved accuracy through feature selection and ensemble learning,’’ Int. J. Adv. Comput. Sci. Appl., vol. 10, no. 1, pp. 252–257, 2019.

[8] A. Aggarwal, A. Rajadesingan, and P. Kumaraguru, ‘‘PhishAri: Automatic realtime phishing detection on Twitter,’’ in Proc. eCrime Res. Summit, Oct. 2012, pp. 1–12.

[9] S. N. Foley, D. Gollmann, and E. Snekkenes, Computer Security— ESORICS 2017, vol. 10492. Oslo, Norway: Springer, Sep. 2017.

[10] P. George and P. Vinod, ‘‘Composite email features for spam identification,’’ in Cyber Security. Singapore: Springer, 2018, pp. 281–289.

[11] H. S. Hota, A. K. Shrivas, and R. Hota, ‘‘An ensemble model for detecting phishing attack with proposed remove-replace feature selection technique,’’ Proc. Comput. Sci., vol. 132, pp. 900–907, Jan. 2018.

[12] G. Sonowal and K. S. Kuppusamy, ‘‘PhiDMA—A phishing detection model with multi-filter approach,’’ J. King Saud Univ., Comput. Inf. Sci., vol. 32, no. 1, pp. 99–112, Jan. 2020.

[13] M. Zouina and B. Outtaj, ‘‘A novel lightweight URL phishing detection system using SVM and similarity index,’’ Hum.-Centric Comput. Inf. Sci., vol. 7, no. 1, p. 17, Jun. 2017.

[14] R. Ø. Skotnes, ‘‘Management commitment and awareness creation—ICT safety and security in electric power supply network companies,’’ Inf. Comput. Secur., vol. 23, no. 3, pp. 302–316, Jul. 2015.

[15] R. Prasad and V. Rohokale, ‘‘Cyber threats and attack overview,’’ in Cyber Security: The Lifeline of Information and Communication Technology. Cham, Switzerland: Springer, 2020, pp. 15–31.

[16] T. Nathezhtha, D. Sangeetha, and V. Vaidehi, ''WC-PAD: Web crawling based phishing attack detection,'' in Proc. Int. Carnahan Conf. Secur. Technol. (ICCST), Oct. 2019, pp. 1–6.

[17] R. Jenni and S. Shankar, ''Review of various methods for phishing detection,'' EAI Endorsed Trans. Energy Web, vol. 5, no. 20, Sep. 2018, Art. no. 155746.

[18] (2020). Accessed: Jan. 2020. [Online]. Available: https://catches-of-themonth-phishing-scams-for-january-2020

[19] S. Bell and P. Komisarczuk, ''An analysis of phishing blacklists: Google safe browsing, OpenPhish, and PhishTank,'' in Proc. Australas. Comput. Sci. Week Multiconf. (ACSW), Melbourne, VIC, Australia. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–11, Art. no. 3, doi: 10.1145/3373017.3373020.

[20] A. K. Jain and B. Gupta, ''PHISH-SAFE: URL features-based phishing detection system using machine learning,'' in Cyber Security. Switzerland: Springer, 2018, pp. 467–474.

[21] Y. Cao, W. Han, and Y. Le, ''Anti-phishing based on automated individual white-list,'' in Proc. 4th ACM Workshop Digit. Identity Manage., Oct. 2008, pp. 51–60.

[22] G. Diksha and J. A. Kumar, ''Mobile phishing attacks and defence mechanisms: State of art and open research challenges,'' Comput. Secur., vol. 73, pp. 519–544, Mar. 2018.

[23] M. Khonji, Y. Iraqi, and A. Jones, ''Phishing detection: A literature survey,'' IEEE Commun. Surveys Tuts., vol. 15, no. 4, pp. 2091–2121, 4th Quart, 2013.

[24] S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs, ''Who falls for phish? A demographic analysis of phishing susceptibility and effectiveness of interventions,'' in Proc. SIGCHI Conf. Hum. Factors Comput. Syst., Apr. 2010, pp. 373–382.

[25] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, ''PhishNet: Predictive blacklisting to detect phishing attacks,'' in Proc. IEEE INFOCOM, Mar. 2010, pp. 1–5.

[26] P. K. Sandhu and S. Singla, ''Google safe browsing-web security,'' in Proc. IJCSET, vol. 5, 2015, pp. 283–287.

[27] M. Sharifi and S. H. Siadati, ''A phishing sites blacklist generator,'' in Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl., Mar. 2008, pp. 840–843.

[28] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, ''An empirical analysis of phishing blacklists,'' in Proc. 6th Conf. Email Anti-Spam (CEAS), Mountain View, CA, USA. Pittsburgh, PA, USA: Carnegie Mellon Univ., Engineering and Public Policy, Jul. 2009.

[29] Y. Zhang, J. I. Hong, and L. F. Cranor, ‘‘Cantina: A content-based approach to detecting phishing web sites,’’ in Proc. 16th Int. Conf. World Wide Web, May 2007, pp. 639–648.

[30] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, ‘‘CANTINA+: A featurerich machine learning framework for detecting phishing web sites,’’ ACM Trans. Inf. Syst. Secur., vol. 14, no. 2, pp. 1–28, Sep. 2011.

[31] C. L. Tan, K. L. Chiew, K. Wong, and S. N. Sze, ‘‘PhishWHO: Phishing webpage detection via identity keywords extraction and target domain name finder,’’ Decis. Support Syst., vol. 88, pp. 18–27, Aug. 2016.

[32] A. Le, A. Markopoulou, and M. Faloutsos, ‘‘PhishDef: URL names say it all,’’ in Proc. IEEE INFOCOM, Apr. 2011, pp. 191–195.

[33] R. Islam and J. Abawajy, ‘‘A multi-tier phishing detection and filtering approach,’’ J. Netw. Comput. Appl., vol. 36, no. 1, pp. 324–335, Jan. 2013.

[34] S. C. Jeeva and E. B. Rajsingh, ‘‘Intelligent phishing URL detection using association rule mining,’’ Hum.-Centric Comput. Inf. Sci., vol. 6, no. 10, pp. 1–19, 2016.

[35] M. Babagoli, M. P. Aghababa, and V. Solouk, ‘‘Heuristic nonlinear regression strategy for detecting phishing websites,’’ Soft Comput., vol. 23, no. 12, pp. 4315–4327, Jun. 2019.

[36] E. Buber, B. Diri, and O. K. Sahingoz, ‘‘Detecting phishing attacks from URL by using NLP techniques,’’ in Proc. Int. Conf. Comput. Sci. Eng. (UBMK), Oct. 2017, pp. 337–342.

[37] E. Buber, B. Diri, and O. K. Sahingoz, ‘‘NLP based phishing attack detection from URLs,’’ in Proc. Int. Conf. Intell. Syst. Design Appl. Cham, Switzerland: Springer, 2017, pp. 608–618.

[38] R. M. Mohammad, F. Thabtah, and L. McCluskey, ‘‘Predicting phishing websites based on self-structuring neural network,’’ Neural Comput. Appl., vol. 25, no. 2, pp. 443–458, Aug. 2014.

[39] F. Feng, Q. Zhou, Z. Shen, X. Yang, L. Han, and J. Wang, ‘‘The application of a novel neural network in the detection of phishing websites,’’ J. Ambient Intell. Hum. Comput., vol. 14, pp. 1–15, Apr. 2018.

[40] S. Smadi, N. Aslam, and L. Zhang, ‘‘Detection of online phishing email using dynamic evolving neural network based on reinforcement learning,’’ Decis. Support Syst., vol. 107, pp. 88–102, Mar. 2018.