iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025

DOI: https://doi.org/10.22214/ijraset.2025.71708

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Ensemble Based Detection of Phishing URLs Using Hybrid, Deep Learning and Machine Learning Models

M. Robin Raj Paul[1], Dr. K. Santhi Sree[2]
[1]*Post Graduate Student, M. Tech(CNIS),* [2]*Professor, Department of Information Technology, Jawaharlal Nehru Technological University, Hyderabad, India*

Abstract:** Phishing attacks pose a serious cybersecurity threat, requiring advanced detection mechanisms. This study proposes an ensemble-based phishing Uniform Resource Locator(URL) detection framework integrating both machine learning and deep learning models. The first phase employs Adaboost, Naïve Bayes(NB), Random Forest(RF), Logistic Regression(LR), Support Vector Machine(SVM), Artificial Neural Network(ANN), Convolutional Neural Network(CNN), Recurrent Neural Network(RNN), Long Short TermMemory(LSTM) and Stacked Gated Recurrent Unit(Stacked GRU), combined using voting ensemble. The second phase includes detection with hybrid deep learning models, including Neural Network -Long Short Term Memory(NN_LSTM), StackedGated Recurrent Unit-Convolutional Neural Network-Long Short Term Memory(StackedGRU_CNN_LSTM), Deep Belief Network -StackedGated Recurrent Unit-Transformer(DBN_StackedGRU_Transformer), Autoencoder+Convolutional Neural Network-Long Short Term Memory+Bi-Gated Recurrent Unit(AutoencoderCNNLSTMBiGRU), and Multi LayerPerceptron-Bi-Long Short Term Memory-Convolutional Neural Network-Gated Recurrent Unit(MLP_BiLSTM_CNN_GRU), utilizing stacking and a host of other ensemble methods like Voting,Weighted Averaging, Confidence-Based Stacking, Gated Mixture of Experts, Neural Greedy Selector, Stacked with Featuresfor improved classification. Performance evaluation using accuracy, precision, recall, and F1-score shows that ensemble learning significantly enhances phishing detection accuracy, making it a robust cybersecurity solution.

Keywords:** Phishing Detection, Machine Learning(ML), Deep Learning(DL), Ensemble Learning, Stacking, Voting, Adaboost(Decision Tree),Naive Bayes, Random Forest, Logistic Regression, Support Vector Machine, Recurrent Neural Network, Artificial Neural Network, Convolutional Neural Network, Long Short Term Memory, Stacked Gated Recurrent Unit, Weighted Averaging, Confidence-Based Stacking, Gated Mixture of Experts, Neural Greedy Selector, Stacked with Features.

I. INTRODUCTION

Phishing is a form of cyberattack where criminals pose as trusted individuals with the goal of tricking victims into sharing sensitive information, including login credentials or financial information. These attacks rely on psychological factors, including trust and time constraints.

Phishing incidents in 2023 totaled almost 5 million, with 3.4 billion phishing messages per day. Phishing has many forms: email phishing, the most prevalent, is when fake emails instruct the recipient to click on dangerous links or download dangerous attachments. Spear phishing addresses specific individuals with tailored information, smishing and vishing address through SMS and voice. Whaling addresses high-profile targets, and pharming leads users to imitation sites.

Phishing URLs are responsible for such attacks. Attackers design fake sites that replicate original sites, often using misspelled domains, alternative extensions, URL shorteners, or even HTTPS to appear legitimate.

Phishing results in financial loss, data breaches, reputational damage, and malware infections. To counter threats in motion, cybersecurity is dependent on machine learning (ML) to detect phishing patterns. Hybrid models derived from ML add security by detecting anomalies and responding to threats in motion.

Phishing strategies, particularly misleading URLs, need to be detected. Powerful cybersecurity solutions, namely ML, need to be employed by individuals and organizations to create pre-emptive defenses against phishing. This paper explores the effectiveness of several machine learning and deep learning models which are ensembled to find out the effectiveness of phishing uniform resource locator(url) detection;along with which hybrid models are also developed to determine their potential of detecting phishing urls.

II. LITERATURE SURVEY

Phishing attacks have become a major cybersecurity concern, prompting extensive research into detection methods. Various approaches, including Rule-Based methods, Machine Learning, Deep Learning and Visual Similarity-based techniques, have been explored to enhance phishing detection. This survey categorizes and discusses key contributions in these areas.

A. Rule-Based and Whitelist-Based Approaches

Rule-based methods detect phishing using predefined patterns. Moghimi and Varjani et al. [7] used custom webpage features in a browser extension, but the manual feature design limits flexibility. Satheesh Kumar et al. [6] proposed an incremental, real-time system analysing URLs, domains, and content, yet its static rules require regular updates.

Whitelist-based methods check sites against trusted domains. Azeez et al. [8] improved detection by comparing visual and actual links to known safe sites, though new or unlisted phishing pages can evade detection.

Overall, both methods are efficient and transparent but struggle to adapt to evolving threats due to their static nature.

B. Machine Learning-Based Approaches

ML techniques classify phishing sites using URL patterns, content, and metadata. Sahingoz et al. [11] used multiple classifiers with NLP features for real-time detection, offering language flexibility but relying on diverse training data. Varshney et al. [3] incorporated lightweight indicators (e.g., HTTPS, Safe Browsing) for efficiency, though these can miss sophisticated phishing sites. Rao and Pais et al. [9] developed Jail-Phish, comparing visual and structural features to detect phishing on compromised servers, but it can be bypassed by minor visual changes.

While ML methods are adaptable and data-driven, they remain sensitive to feature quality, training data limitations, and adversarial tactics.

C. Deep Learning-Based Approaches

Recent deep learning advancements have greatly enhanced phishing detection by automating feature extraction and improving pattern recognition. These models, with their hierarchical learning capabilities, outperform traditional methods in identifying complex phishing patterns in URLs and websites.

Sahingoz et al. [1] proposed DEPHIDES, a deep learning-based phishing detection system, evaluating various neural networks like ANNs, CNNs, RNNs, BiRNNs, and attention-based models. Their experiments, conducted on a large-scale dataset of millions of URLs, highlighted the effectiveness of CNNs for real-time cybersecurity applications.

Huang et al. [10] introduced a model combining CNNs and hierarchical attention-based RNNs for phishing URL classification, demonstrating the model’s ability to outperform others like LSTM–CNN and standalone LSTM approaches by leveraging both spatial and sequential characteristics of URLs.

Singh et al. [12] developed a deep learning framework using CNNs, LSTMs, and CNN-LSTM hybrids for phishing URL classification without the need for manual feature engineering. Their findings reinforced the potential of CNN-based approaches for end-to-end phishing detection.

Asiri et al. [13] introduced PhishingRTDS, a real-time phishing detection system using BiLSTM networks with attention mechanisms, providing an efficient solution for detecting evolving phishing threats with minimal latency.

Majgave and Gavankar et al. [14] proposed the Transformer-Based Deep Belief Network (TB-DBN), which integrates transfer learning, transformers, and autoencoders to address issues like data imbalance and improve generalization, reducing the need for handcrafted features.

While these approaches show great promise, challenges such as extensive computational requirements, long training times, and vulnerability to adversarial attacks remain. Additionally, large labeled datasets and appropriate regularization are necessary to prevent overfitting.

D. Visual Similarity-Based Approaches

To counter increasingly deceptive phishing tactics, visual similarity-based techniques focus on how closely a suspicious website resembles a legitimate one in appearance. These methods offer an alternative to traditional approaches that rely on analysing URLs or HTML content. Zhou et al. [2] introduced a strategy that examines both localized elements, like logos, and broader structural features of a webpage. By combining these components, their system enhances the accuracy of phishing site identification.

Liu et al. [4] presented *SiteWatcher*, a system that first scans emails for potential threats and then visually evaluates the suspect webpages against authentic ones. It identifies phishing attempts by analysing visual aspects such as page layout, design patterns, and key interface areas.

Medvet et al. [5] proposed a framework that compares textual elements, images, and visual structure to detect fraudulent sites. Their multi-faceted approach helps improve detection effectiveness by leveraging various visual cues.

However, these methods often depend on comprehensive databases of trusted websites for comparison. Additionally, they can face difficulties when dealing with dynamic content or websites that frequently change their appearance, which may impact detection consistency.

E. Hybrid Approaches

Hybrid phishing detection methods combine machine learning and deep learning techniques to leverage their strengths, improving detection by capturing diverse phishing patterns. Sahingoz et al. [1] developed a system using CNNs, RNNs, Bi-RNNs, and attention mechanisms, focusing on fast URL-based webpage classification for large-scale cybersecurity applications.

Huang et al. [10] proposed a model combining CNNs for feature extraction with attention-based hierarchical RNNs, enabling the system to effectively capture both spatial and temporal features of URLs.

Asiri et al. [13] introduced a real-time detection framework using bidirectional LSTMs with attention layers, which dynamically focuses on key input parts, enhancing timely and accurate phishing detection.

While hybrid models improve detection, they introduce computational overhead and complexity, which may limit their use in resource-constrained environments.

Therefore ,building on this existing research this paper explores the effectiveness of individual machine learning and deep learning models and their ensemble in detecting phishing urls and hybrid models are evaluated to determine their detection efficiency.

III. PROPOSED WORK

This paper proposes a multi-level ensemble-based phishing URL detection system that integrates traditional machine learning models with advanced hybrid deep learning architectures. Initially, baseline ML and DL models are combined using soft voting to establish a performance benchmark. The core contribution is the design of five hybrid neural models—incorporating BiLSTM, CNN, GRU, Autoencoders, Transformers, and attention layers—which are further fused using adaptive ensemble strategies like Confidence Stacking, Gated Mixture of Experts and several others. This approach enhances detection accuracy by capturing complex URL patterns and dynamically leveraging the strengths of individual models.

A. Dataset Description

The dataset used in this project is a comprehensive phishing detection dataset containing 88,647 entries and 112 featureset al.[20]. It is designed to analyse various URL characteristics that help distinguish phishing websites from legitimate ones. The dataset includes structural, domain-based, directory and file-based, parameter-relatedand security-related features. These features allow for training and evaluation of machine learning models to enhance phishing detection capabilities.

Key features in the dataset include:

1) qty_dot_url: Number of dots in the URL.
2) qty_hyphen_url: Number of hyphens in the URL.
3) qty_slash_url: Number of slashes in the URL.
4) qty_questionmark_url: Presence of a question mark in the URL.
5) qty_at_url: Number of '@' symbols in the URL.
6) domain_length: Length of the domain name.
7) tls_ssl_certificate: Indicates whether the website has an SSL certificate (1 for Yes, 0 for No).
8) url_shortened: Identifies if the URL uses a URL-shortening service.
9) phishing: The target variable (1 for phishing, 0 for legitimate).

qty_dot_url	qty_hyphen_url	qty_slash_url	qty_questionmark_url	qty_at_url	domain_length	tls_ssl_certificate	url_shortened	Phishing
3	0	1	0	0	23	0	0	1

5	0	3	0	0	28	1	0	1
2	0	1	0	0	17	1	0	0
4	0	5	0	0	26	1	0	1
2	0	0	0	0	19	0	0	0

Table-1:Dataset Description

B. Architecture

The General Architecture of this project is as mentioned below,it is later divided into two phases for execution purposes.

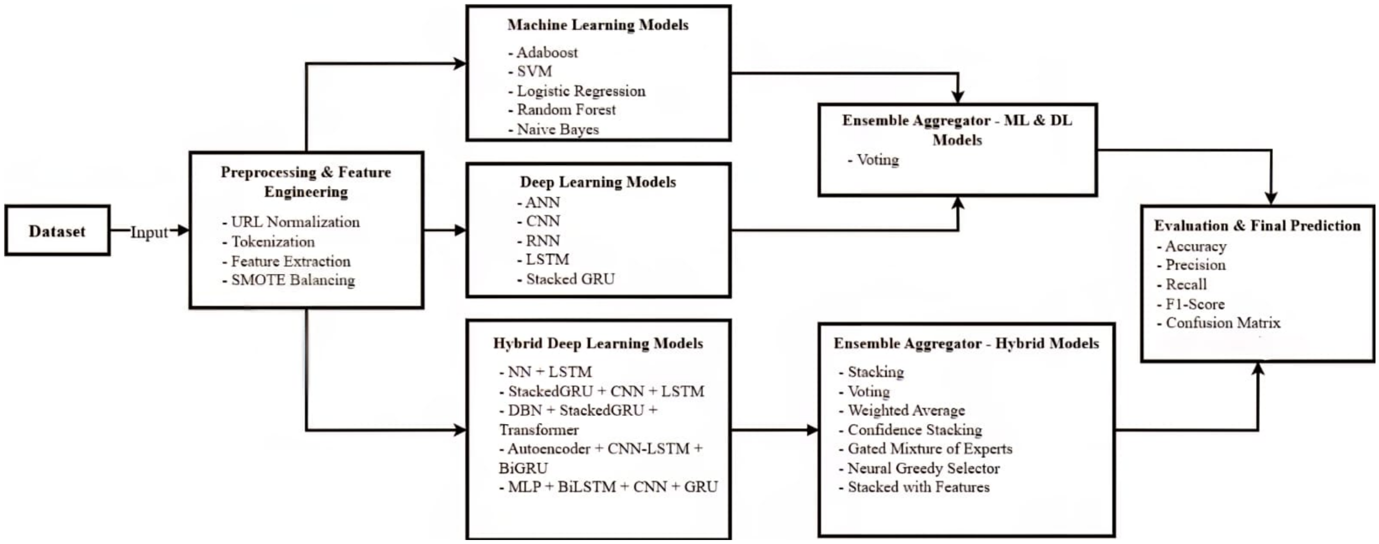

Figure-1: General Architecture

The Models used in order to carry out this paper can be clearly classified into two groups which have been executed in two phases and the models are listed below

Group-1 (Phase-1)	Group-2 (Phase-2)
Adaboost(Decision Tree) Naive Bayes Random Forest Logistic Regression Support Vector Machine Recurrent Neural Network Artificial Neural Network Convolutional Neural Network Long Short TermMemory Stacked Gated Recurrent Unit	Neural Network + Long Short-Term Memory (NN_LSTM) Stacked Gated Recurrent Unit with Convolutional Neural Network and Long Short-Term Memory(StackedGRU_CNN_LSTM) Deep Belief Network with Stacked Gated Recurrent Unit and Transformer(DBN_StackedGRU_Transformer) Autoencoder with Convolutional Neural Network, Long Short-Term Memory, and Bidirectional Gated Recurrent Unit(AutoencoderCNNLSTMBiGRU) Multilayer Perceptron with Bidirectional Long Short-Term Memory, Convolutional Neural Network, and Gated Recurrent Unit(MLP_BiLSTM_CNN_GRU)

Table-2:Model Description

For the Group-1 all of the models mentioned are the base learners for Soft Voting Ensemble;Similarly for Group-2 these hybrid neural architectures are the base learners for the following Ensemble Methods:

1) Stacked with Features

2) Gated Mixture of Experts
3) Confidence Stacking
4) Weighted Average
5) Neural Greedy Selector
6) Stacking
7) Voting

This paper is implemented in two phases and in the first phase it contains all of the machine learning and deep learning models whose individual performance is evaluated and then their predictions are passed through Voting Ensemble in order to achieve the best performance; the architecture for the same is as depicted below.

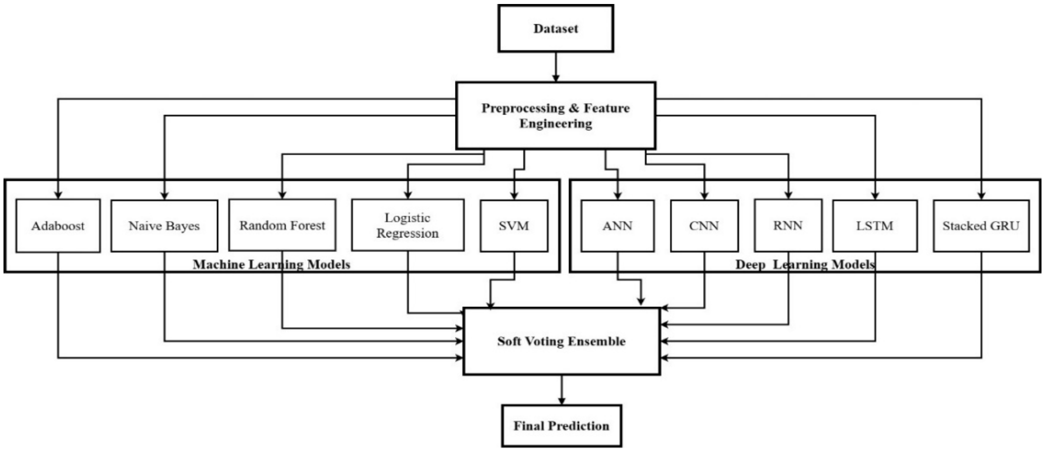

Figure-2: Phase-1 Architecture

Phase-2 of the project involves dealing with hybrid neural networks which are again individually evaluated and then passed through different ensemble methods in order to establish the fact that these hybrid neural networks perform better than the individual models and the ensemble methods can even exceed the performance of some hybrid models, the architecture for the same is as depicted below

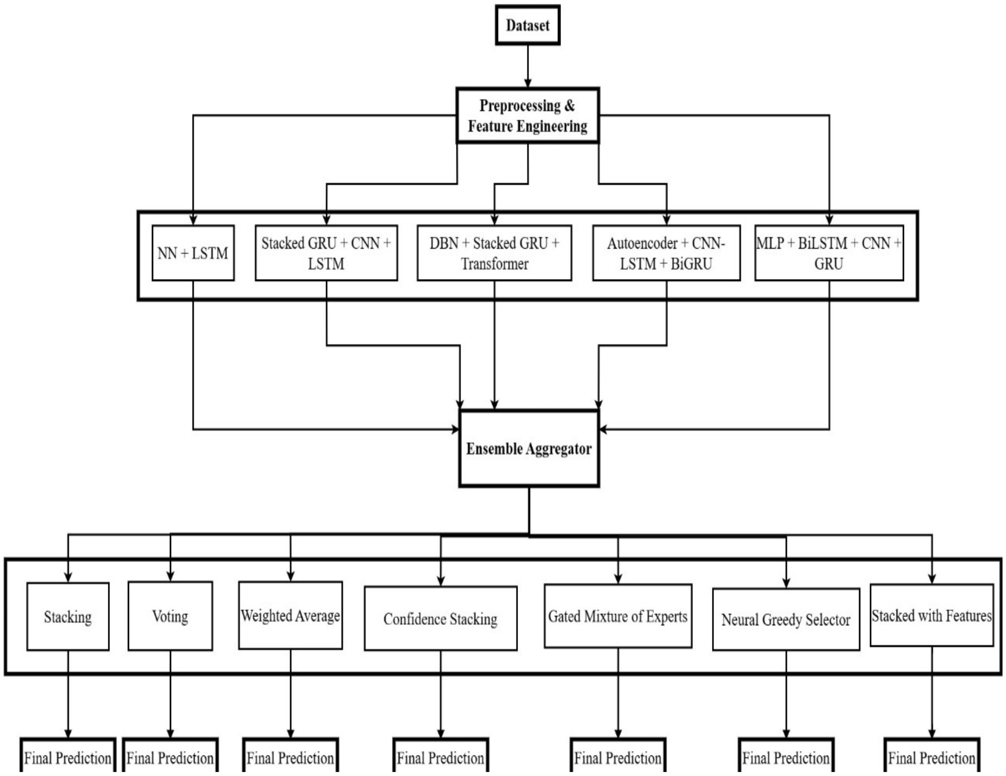

Figure-3: Phase-2 Architecture

C. Methodology

The entire project methodology is as follows:

1) Input Dataset

The project starts with thedataset containing feature-engineered data derived from URLs. These features include structural properties, lexical characteristics, and statistical indicators. The dataset is labelled, indicating whether each URL is phishing (label 1) or legitimate (label 0).

2) Preprocessing and Feature Engineering

Before model training, the dataset undergoes several preprocessing steps:

- URL normalization (e.g., lowercasing, parameter removal)
- Tokenization of URLs into parts such as subdomain, domain, and path
- Feature extraction based on patterns, entropy, length, and character distributions
- Handling class imbalance using SMOTE (Synthetic Minority Oversampling Technique)
- Standardization or normalization of feature vectors

The resulting dataset is balanced and transformed into numerical feature vectors ready for model consumption.

3) Model Training Categories

The models are grouped into three main categories and trained independently:

Category-1:Traditional Machine Learning Models

These include:

- Adaboost
- Support Vector Machine (SVM)
- Logistic Regression
- Random Forest
- Naive Bayes

These models operate on manually extracted features and use classical training paradigms.

Category-2:Deep Learning Models

These models automatically learn hierarchical and sequential patterns:

- Artificial Neural Network (ANN)
- Convolutional Neural Network (CNN)
- Recurrent Neural Network (RNN)
- Long Short-Term Memory (LSTM)
- Stacked Gated Recurrent Unit (Stacked GRU)

Category-3:Hybrid Deep Learning Models

These are advanced, custom-designed architectures combining multiple components to capture spatial, temporal, and contextual URL features. The Hybrid Models are:

- NN + LSTM
- StackedGRU + CNN + LSTM
- DBN + StackedGRU + Transformer
- Autoencoder + CNN-LSTM + Bi-GRU
- MLP + BiLSTM + CNN + GRU

These models are designed to capture long-term dependencies, compressed representations, and deep feature interactions.

4) Ensemble Aggregation

After training individual models, their predictions are aggregated using ensemble learning techniques to boost overall accuracy and generalization.

Ensemble of ML and DL Models:

Soft Voting: Aggregates probability scores from ML and DL models to make a final decision.

Ensemble of Hybrid Deep Learning Models:

Multiple ensemble strategies are applied such as Stacking, Voting, Weighted Averaging, Confidence-Based Stacking, Gated Mixture of Experts, Neural Greedy Selector, Stacked with Features

5) Evaluation

Each model and ensemble method is evaluated using standard classification metrics like Accuracy, Precision, Recall, F-Score, Confusion Matrix, Classification Report

To Summarise the major steps involved in the project is as follows:

- Load and preprocess the dataset.
- Apply SMOTE to balance the classes.
- Train individual ML, DL, and hybrid models.
- Store and collect predictions.
- Apply ensemble aggregation techniques.
- Evaluate and compare all models and ensembles.

IV. EXPERIMENTAL ANALYSIS AND RESULTS

This project has been executed as mentioned in the above stages in two phases i.e. Individual Models and Hybrid Models respectively.The results of the implementation are as follows and the hybrid models with ensemble clearly outperforms other models.

A. Performance Metrics

Phase-1 Execution Results:

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
AdaBoost	0.9242	0.8564	0.9380	0.8954	0.9783
Random Forest	0.9694	0.9473	0.9651	0.9561	0.9947
Logistic Regression	0.9220	0.8492	0.9416	0.8930	0.9792
SVM	0.9213	0.8449	0.9458	0.8925	0.9791
Naive Bayes	0.7812	0.8845	0.4222	0.5716	0.9548
ANN	0.9571	0.9240	0.9545	0.9390	0.9911
CNN	0.9560	0.9234	0.9517	0.9373	0.9913
RNN	0.9390	0.8780	0.9564	0.9155	0.9883
LSTM	0.9584	0.9279	0.9537	0.9406	0.9911
Stacked GRU	0.9582	0.9317	0.9486	0.9401	0.9908
Voting Classifier	0.9569	0.9233	0.9545	0.9386	0.9911

Table-3: Phase-1 Results

Phase-2 Execution Results

Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
NN_LSTM	0.9603	0.9340	0.9527	0.9432	0.9921

Model	Accuracy	Precision	Recall	F1-Score	ROC AUC
StackedGRU_CNN_LSTM	0.9589	0.9323	0.9502	0.9412	0.9913
DBN_StackedGRU_Transformer	0.9559	0.9358	0.9367	0.9362	0.9905
AutoencoderCNNLSTMBiGRU	0.9545	0.9215	0.9493	0.9352	0.9903
MLP_BiLSTM_CNN_GRU	0.9600	0.9329	0.9527	0.9427	0.9917
Voting Classifier	0.9613	0.9295	0.9608	0.9449	0.9927
Stacking Classifier	0.9638	0.9388	0.9577	0.9482	0.9925
StackedWithFeatures	0.9614	0.9337	0.9561	0.9448	0.9922
GatedMixtureOfExperts	0.9618	0.9359	0.9550	0.9453	0.9921
ConfidenceStacking	0.9608	0.9287	0.9604	0.9443	0.9920
Weighted Average	0.9619	0.9349	0.9564	0.9456	0.9920
Neural Greedy Selector	0.9621	0.9403	0.9507	0.9455	0.9921

Table-4: Phase-2 Results

Therefore this project successfully demonstrates that ensemble techniques significantly enhance detection performance. Amongst all models, advanced ensemble strategies—especially stacking and neural fusion—consistently outperformed individual Machine Learning and Deep Learning models across all evaluation metrics. These results affirm that ensemble learning offers a robust, accurate, and reliable solution for phishing URL detection in real-world cybersecurity applications.

B. Plots and Visualizations

Confusion Matrices and Learning Curves have been plotted to get an understanding of the models and their performances.

Phase-1 Plots are as follows:

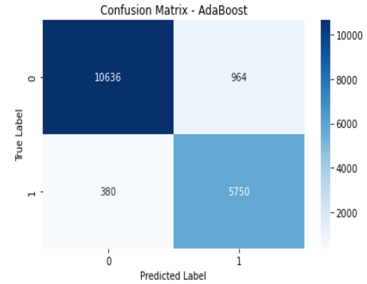

Figure-4: Adaboost Confusion Matrix

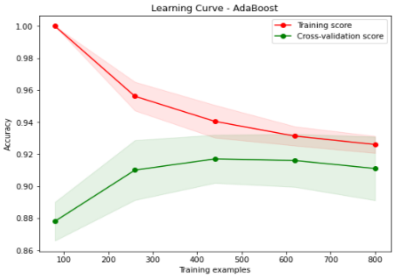

Figure-5: Adaboost Learning Curves

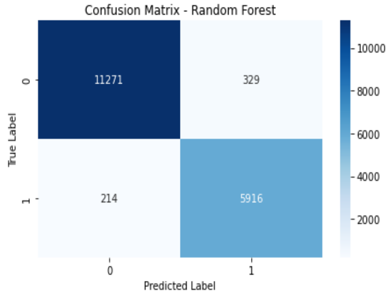

Figure-6: Random Forest Confusion Matrix

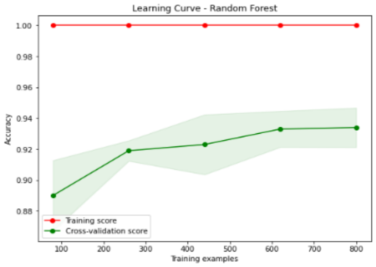

Figure-7: Random Forest Learning Curves

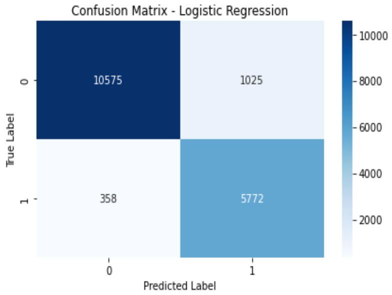

Figure-8: Logistic Regression Confusion Matrix

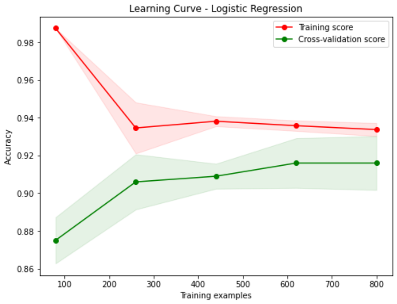

Figure-9: Logistic Regression Learning Curves

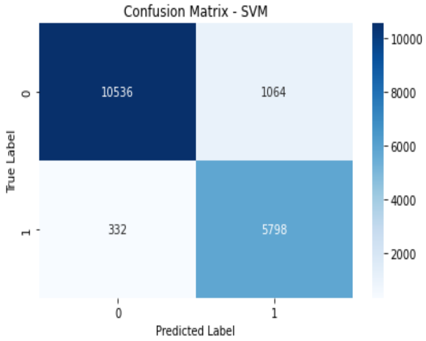

Figure-10: SVM Confusion Matrix

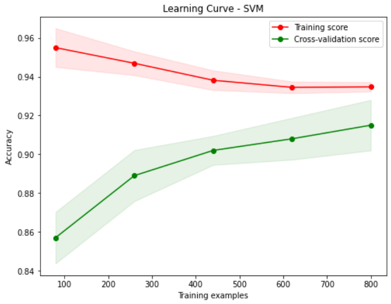

Figure-11:SVM Learning Curves

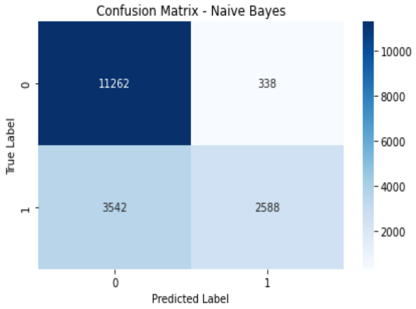

Figure-12: Naïve Bayes Confusion Matrix

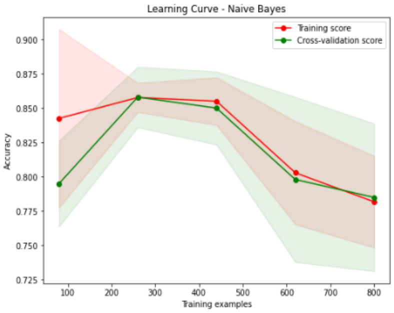

Figure-13: Naïve Bayes Learning Curves

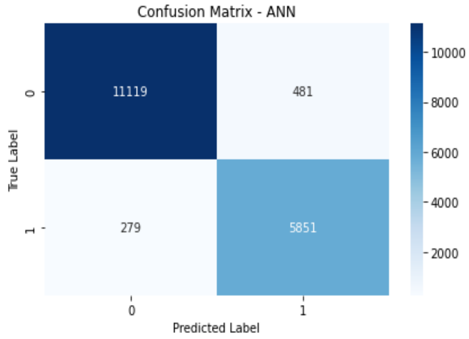

Figure-14: ANN Confusion Matrix

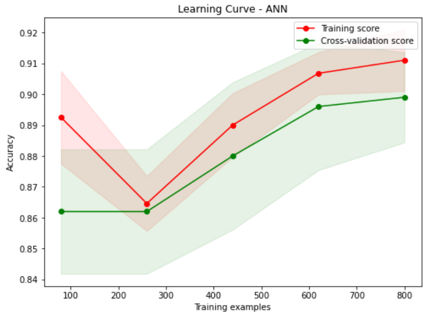

Figure-15:ANN Learning Curves

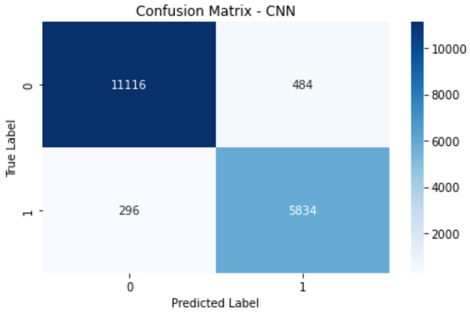

Figure-16: CNN Confusion Matrix

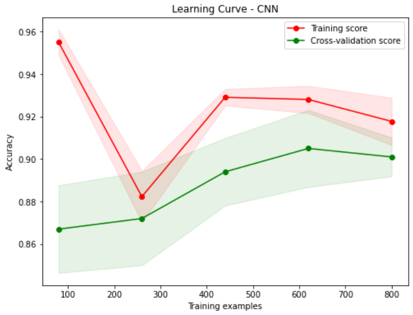

Figure-17:CNN Learning Curves

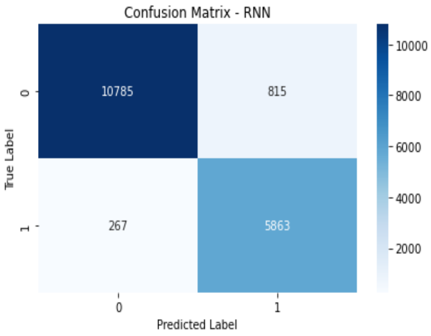

Figure-18: RNN Confusion Matrix

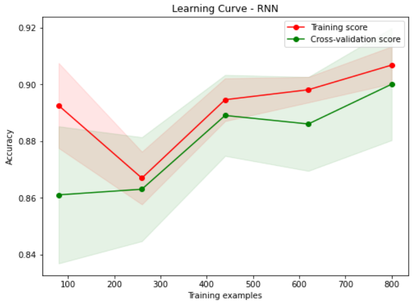

Figure-19: RNN Learning Curves

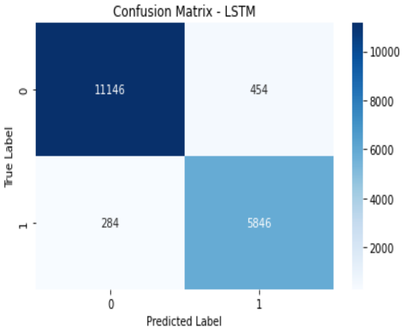

Figure-20: LSTM Confusion Matrix

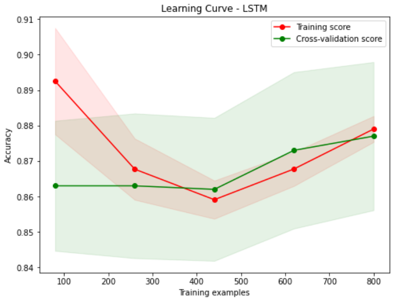

Figure-21: LSTM Learning Curves

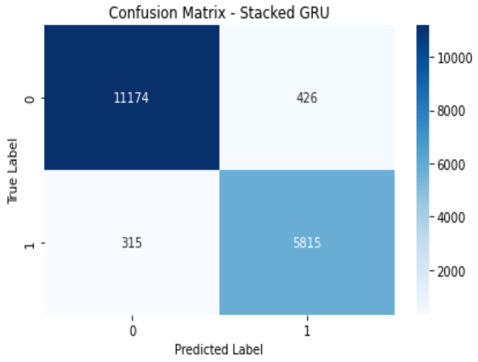

Figure-22:Stacked GRU Confusion Matrix

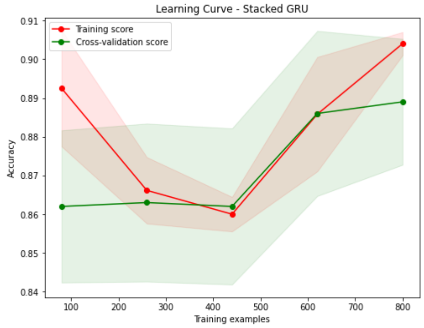

Figure-23:Stacked GRU Learning Curves

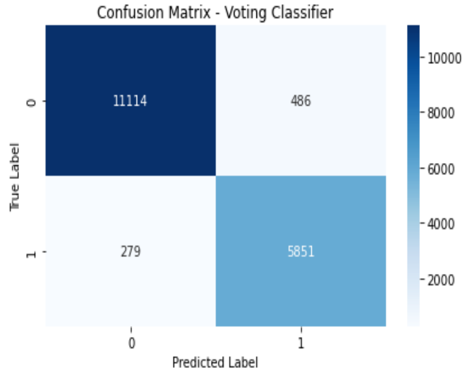

Figure-24: Voting Classifier Confusion Matrix

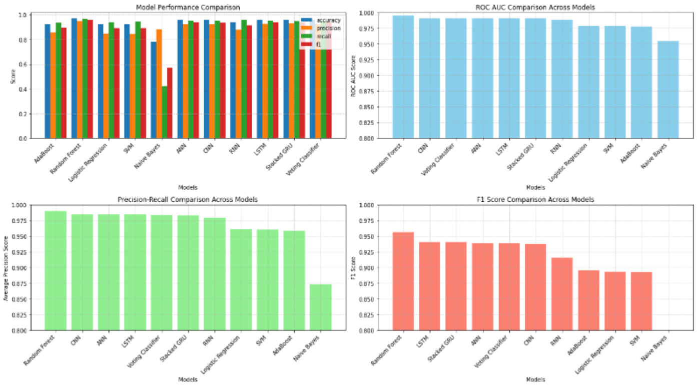

Figure-25:Model Performance Plots

Phase-2 Plots are as follows:

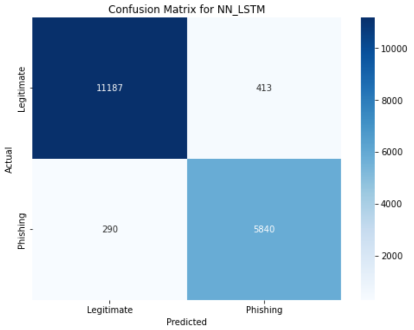

Figure-26: NN_LSTM Confusion Matrix

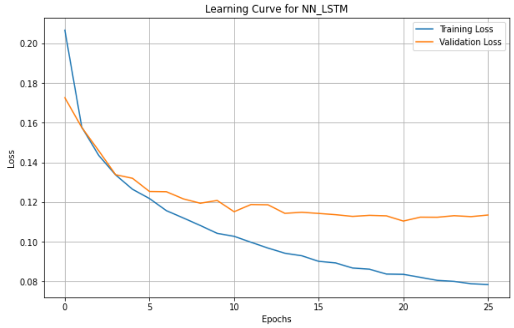

Figure-27: NN_LSTM Learning Curves

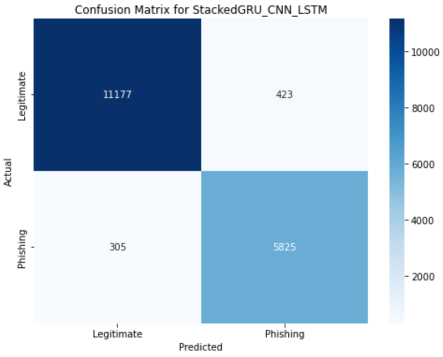

Figure-28: StackedGRU_CNN_LSTM Confusion Matrix

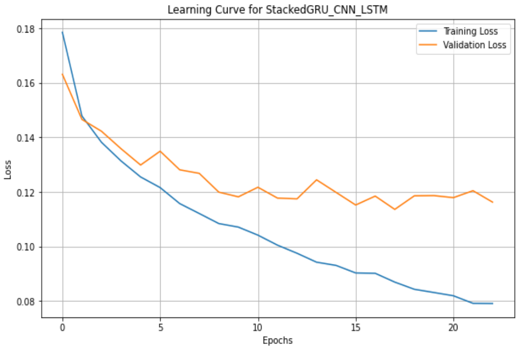

Figure-29: StackedGRU_CNN_LSTM Learning Curves

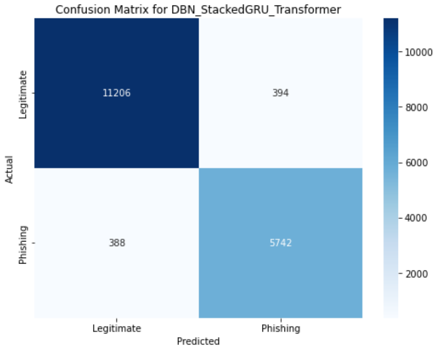

Figure-30: DBNStackedGRUTransformer Confusion Matrix

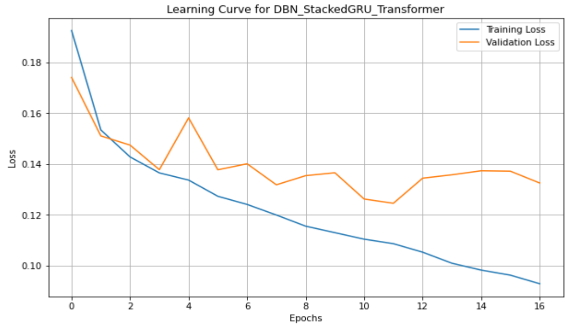

Figure-31: DBN_StackedGRU_Transformer Learning Curves

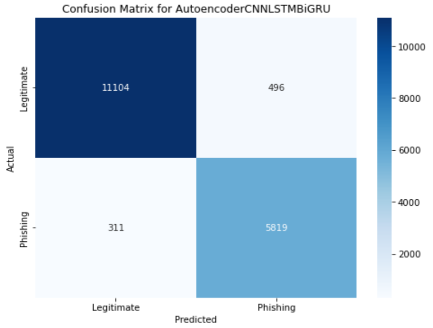

Figure-32: AutoencoderCNNLSTMBiGRU Confusion Matrix

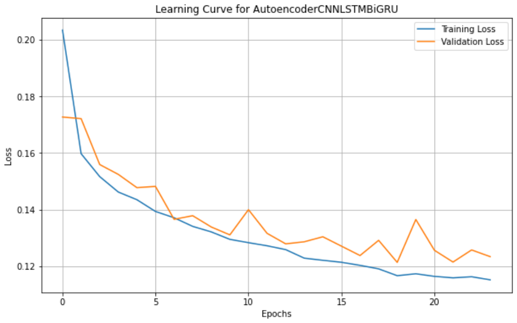

Figure-33: Autoencoder_CNN_LSTM_BiGRU Learning Curves

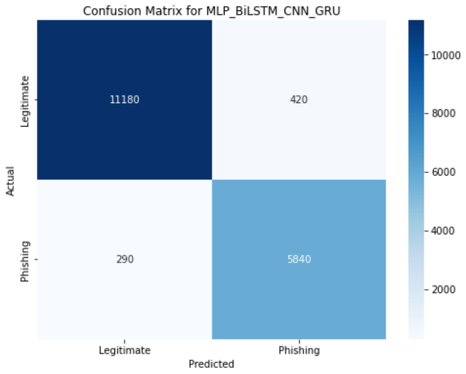

Figure-34: MLP_BiLSTM_CNN_GRU Confusion Matrix

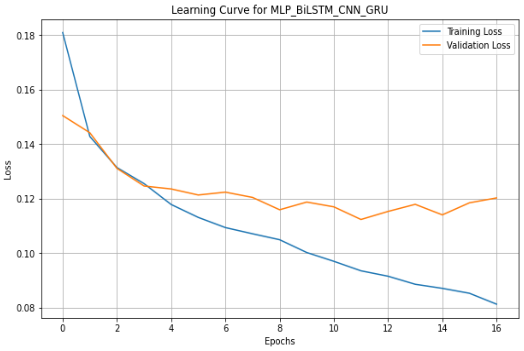

Figure-35: MLP_BiLSTM_CNN_GRU Learning Curves

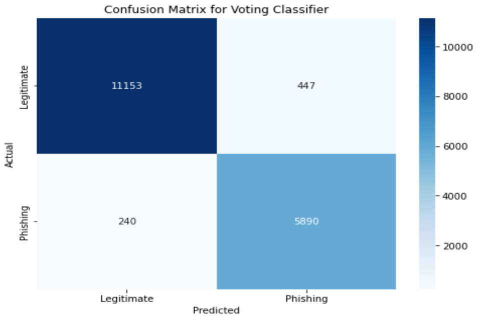

Figure-36: Voting Confusion Matrix

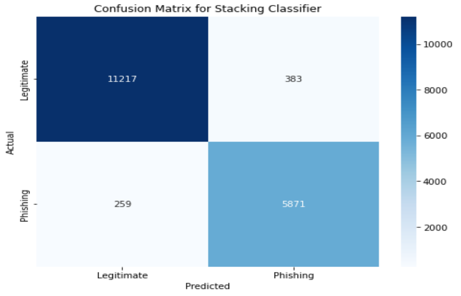

Figure-37:Stacking Confusion Matrix

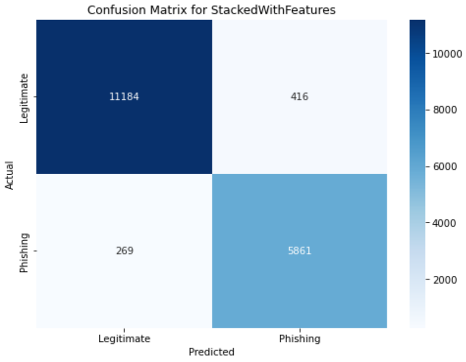

Figure-38: Stacked with Features Confusion Matrix

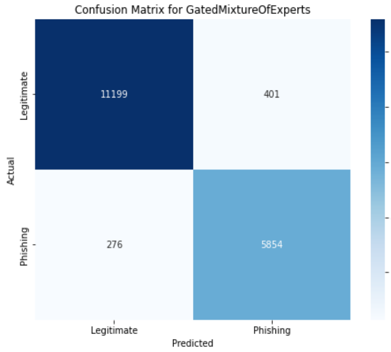

Figure-39: Gated Mixture of Experts Confusion Matrix

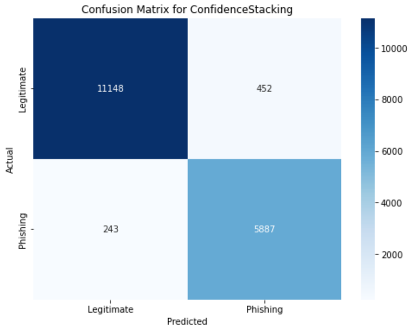

Figure-40: Confidence Stacking Confusion Matrix

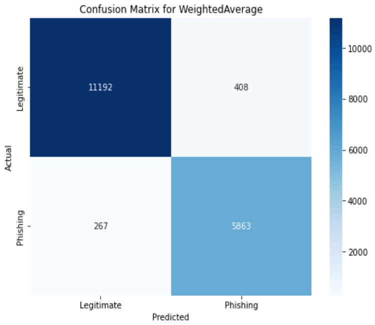

Figure-41: Weighted Average Confusion Matrix

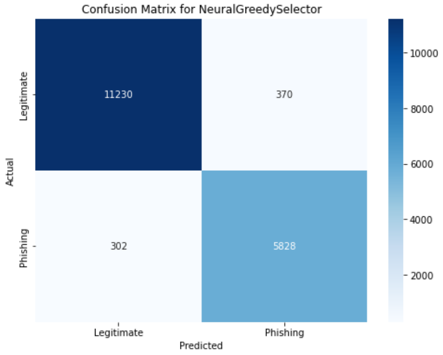

Figure-42: Neural Greedy Selector Confusion Matrix

The visual analysis of both Phase-1 and Phase-2 models demonstrates a clear progression in detection capability as the methodology evolves from standalone machine and deep learning models to more complex hybrid and ensemble architectures. In Phase-1, while traditional classifiers like Random Forest and Logistic Regression showed competitive baseline performance, deep learning models such as LSTM and Stacked GRU captured sequential patterns more effectively. However, limitations in consistency and precision were evident in several individual models. In contrast, Phase-2 results highlight significant improvements through hybrid architectures that combine multiple learning mechanisms—such as CNNs, LSTMs, GRUs, and Transformers—alongside attention mechanisms and feature fusion strategies. Ensemble techniques, particularly those leveraging confidence stacking, gating, and dynamic selection, further enhanced performance by integrating model strengths and minimizing their weaknesses. The ROC curves, confusion matrices, and precision-recall plots collectively affirm that hybrid ensembles offer not only higher predictive accuracy but also better generalization and robustness against diverse phishing patterns. These findings underscore the effectiveness of an integrated approachi.e. ensembles in addressing the complexities of modern phishing threats.

V. CONCLUSIONS

This study highlights the superior effectiveness of ensemble-based learning in phishing URL detection by integrating both classical machine learning models and advanced hybrid deep learning architectures. Among the individual models evaluated, Random Forest from Phase-1 and NN_LSTM from Phase-2 delivered the most competitive standalone performances, with high accuracy and ROC AUC scores. However, all ensemble strategies consistently outperformed individual models across all evaluation metrics.The best-performing model across both phases was the Stacking Classifier applied to hybrid deep learning models, achieving the highest accuracy (96.38%), precision (93.88%), recall (95.77%), F1-score (94.82%), and ROC AUC (0.9925). This clearly demonstrates that the strategic fusion of diverse neural architectures through advanced ensembling techniques leads to significantly improved generalization and robustness.

Furthermore, methods such as Weighted Average, Gated Mixture of Experts, and Neural Greedy Selector closely followed the top performer, reinforcing the conclusion that ensemble frameworks—especially those integrating attention-aware and meta-learning components—provide a powerful, scalable, and dependable solution for phishing detection. These findings affirm that combining heterogeneous models in a thoughtfully constructed ensemble is essential for addressing the complexities of modern phishing attacks.

REFERENCES

[1] O. Sahingoz, E. Buber, and E. Kugu, "DEPHIDES: Deep Learning Based Phishing Detection System," IEEE Access, pp. 1–1, 2024. doi: 10.1109/ACCESS.2024.3352629.

[2] Y. Zhou, Y. Zhang, J. Xiao, Y. Wang, and W. Lin, "Visual Similarity Based Anti-phishing with the Combination of Local and Global Features," in 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, Beijing, China, 2014, pp. 189–196. doi: 10.1109/TrustCom.2014.28.

[3] G. Varshney, M. Misra, and P. K. Atrey, "Improving the accuracy of Search Engine based anti-phishing solutions using lightweight features," in 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST), Barcelona, Spain, 2016, pp. 365–370. doi: 10.1109/ICITST.2016.7856731.

[4] W. Liu, X. Deng, G. Huang, and A. Y. Fu, "An antiphishing strategy based on visual similarity assessment," IEEE Internet Computing, vol. 10, no. 2, pp. 58–65, Mar.–Apr. 2006. doi: 10.1109/MIC.2006.23.

[5] E. Medvet, E. Kirda, and C. Kruegel, "Visual-similarity-based phishing detection," in Proc. 4th Int. Conf. Security Privacy Commun. Netw. (SecureComm '08), New York, NY, USA, 2008, pp. 1–6. doi: 10.1145/1460877.1460905.

[6] M. SatheeshKumar, K. G. Srinivasagan, and G. UnniKrishnan, "A lightweight and proactive rule-based incremental construction approach to detect phishing scam," Inf. Technol. Manag., vol. 23, no. 4, pp. 271–298, Dec. 2022. doi: 10.1007/s10799-021-00351-7.

[7] M. Moghimi and A. Y. Varjani, "New rule-based phishing detection method," Expert Syst. Appl., vol. 53, pp. 231–242, 2016. doi: 10.1016/j.eswa.2016.01.028.

[8] N. A. Azeez, S. Misra, I. A. Margaret, L. Fernandez-Sanz, and S. M. Abdulhamid, "Adopting automated whitelist approach for detecting phishing attacks," Comput. Secur., vol. 108, Sep. 2021. doi: 10.1016/j.cose.2021.102328.

[9] R. S. Rao and A. R. Pais, "Jail-Phish: An improved search engine based phishing detection system," Comput. Secur., vol. 83, pp. 246–267, Jun. 2019. doi: 10.1016/j.cose.2019.02.011.

[10] Y. Huang, Q. Yang, J. Qin, and W. Wen, "Phishing URL Detection via CNN and Attention-Based Hierarchical RNN," in 2019 18th IEEE Int. Conf. Trust, Security and Privacy in Comput. Commun. (TrustCom/BigDataSE), Rotorua, New Zealand, 2019, pp. 112–119. doi: 10.1109/TrustCom/BigDataSE.2019.00024.

[11] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," Expert Syst. Appl., vol. 117, pp. 345–357, 2019. doi: 10.1016/j.eswa.2018.09.029.

[12] S. Singh, M. P. Singh, and R. Pandey, "Phishing Detection from URLs Using Deep Learning Approach," in 2020 5th Int. Conf. Comput., Commun. Security (ICCCS), Patna, India, 2020, pp. 1–4. doi: 10.1109/ICCCS49678.2020.9277459.

[13] S. Asiri, Y. Xiao, S. Alzahrani, and T. Li, "PhishingRTDS: A real-time detection system for phishing attacks using a Deep Learning model," Comput. Secur., vol. 141, 2024. doi: 10.1016/j.cose.2024.103843.

[14] A. B. Majgave and N. L. Gavankar, "Automatic phishing website detection and prevention model using transformer deep belief network," Comput. Secur., vol. 147, 2024. doi: 10.1016/j.cose.2024.104071.

[15] Anti-Phishing Working Group, "Phishing Attacks Trends Report-Q2 2022," Sep. 2022. Accessed: Oct. 15, 2022. [Online]. Available: https://apwg.org/trendsreports/

[16] Cloudflare, "2023 Phishing Threats Report," Oct. 1, 2023. Accessed: [Online]. Available: https://www.cloudflare.com/lp/2023-phishing-report/

[17] M. Volkamer, K. Renaud, B. Reinheimer, and A. Kunz, "User experiences of TORPEDO: Tooltip-powered phishing email detection," Comput. Secur., vol. 71, pp. 100–113, Nov. 2017. doi: 10.1016/j.cose.2017.02.004.

[18] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, "Deep learning for phishing detection: Taxonomy, current challenges and future directions," IEEE Access, vol. 10, pp. 36429–36463, 2022. doi: 10.1109/ACCESS.2022.3151903.

[19] T. Mahara, V. L. H. Josephine, R. Srinivasan, P. Prakash, A. D. Algarni, and O. P. Verma, "Deep vs. shallow: A comparative study of machine learning and deep learning approaches for fake health news detection," IEEE Access, vol. 11, pp. 79330–79340, 2023. doi: 10.1109/ACCESS.2023.3298441.

[20] G. Vrbančič, "Phishing Websites Dataset," Mendeley Data, V1, 2020. doi: 10.17632/72ptz43s9v.1.

[21] N. Abdelhamid, A. Ayesh, and F. Thabtah, "Phishing detection based associative classification data mining," Expert Syst. Appl., vol. 41, no. 13, pp. 5948–5959, Oct. 2014. doi: 10.1016/j.eswa.2014.03.019