

Enhancing Cybersecurity Against Phishing Attacks Using VGA-Hyphishnet: A Hybrid Generative–Discriminative Model

K. Sravanthi* and N. Syed Siraj
Ahmed

School of Information Science
Presidency University, Bengaluru
560064, Karnataka, India

*E-mail: sravanthi.20233CSC0010@
presidencyuniversity.in

Received for publication
May 09, 2025.

Abstract

Traditional phishing detection systems based on machine learning (ML) and deep learning (DL) techniques largely rely on static patterns and manually engineered features. However, as phishing attacks continue to evolve in complexity and sophistication, these conventional approaches face significant limitations in accurately identifying emerging threats. To address this challenge, this study proposes VGA-HyPhishNet, a self-evolving generative artificial intelligence (AI) framework designed for real-time phishing URL detection. The proposed architecture integrates a generative encoder, variational generative modeling (VGM), a self-evolving generative threat engine (SE-GTE), and a hybrid generative–discriminative classifier to learn distribution-aware latent representations directly from raw URL strings. Unlike traditional discriminative models that rely solely on labeled patterns, the proposed framework leverages synthetic latent threat generation to dynamically adapt to evolving phishing behaviors and probabilistic structural patterns. The effectiveness of the proposed model was evaluated using the URL-Phish dataset, consisting of 111,660 URLs. Experimental results demonstrate that VGA-HyPhishNet achieves an accuracy of 98.73%, precision of 93.00%, recall of 94.97%, and an F1-score of 93.97%. Furthermore, comparative analysis with baseline models, including URL-transformer, BiLSTM-URL, and CharCNN-URL, shows that the proposed framework significantly reduces false positives (FP) and false negatives (FN) while maintaining low inference latency suitable for real-time deployment. Overall, the results indicate that the proposed work provides a robust and reliable solution for modern cybersecurity systems by leveraging generative representation learning and adaptive threat modeling, thereby improving resilience against zero-day phishing attacks and adversarial threats.

Keywords

cybersecurity, deep learning, generative AI, hybrid learning, latent representation learning, phishing detection, self-evolving threat modeling, real-time detection, URL classification, variational modeling

1. Introduction

Phishing is a form of social engineering attack that targets users' personal and financial information by deceiving them into revealing sensitive data such as

login credentials, banking details, and identity information [1]. One of the most common techniques used in phishing attacks is email spoofing, where attackers generate emails with falsified sender addresses to impersonate legitimate organizations or trusted

entities. These phishing emails often appear to originate from reputable sources and are designed to lure users into visiting malicious websites or submitting confidential information through fraudulent forms. In many cases, cybercriminals also deploy malware on compromised systems to steal user credentials and gain unauthorized access to online accounts [2].

Phishers exploit multiple communication channels to collect user data, including email, online forums, URLs, instant messaging (IM), SMS, and phone calls [3]. The structure and appearance of phishing content are deliberately crafted to resemble authentic communications, thereby encouraging users to trust the message and unknowingly disclose sensitive information [4]. The primary objective of phishing attacks is to gain unauthorized access to confidential information for financial fraud or identity theft. As a result, phishing attacks have caused significant damage to individuals, organizations, and financial institutions worldwide [5].

The anti-phishing working group (APWG), a non-profit organization, maintains a global repository of verified phishing URLs and regularly publishes reports on phishing activities across the world [6]. According to recent reports, phishing incidents have increased dramatically in recent years, with the number of attacks nearly doubling between 2021 and 2022. In July 2022 alone, more than 280,000 phishing attempts were reported globally [7]. Webmail services remain one of the most frequently targeted sectors, and phishing attempts against well-known organizations increased from approximately 500 incidents in July 2022 to nearly 900 incidents by November 2022. Such attacks not only cause financial losses but also lead to severe cybersecurity risks for organizations and individuals.

A typical example of a phishing cyberattack involves sending an email that appears to originate from a trusted organization, such as a bank or corporate service provider, with the intention of persuading the recipient to open a malicious attachment or click on a fraudulent link. Once the victim interacts with the malicious content, attackers may gain access to confidential user information. Furthermore, phishing emails and websites are often carefully designed to closely mimic legitimate business platforms, making them difficult for users to distinguish from genuine services [8].

To address these challenges, artificial intelligence (AI) technologies have rapidly evolved and introduced intelligent techniques such as machine learning (ML) and deep learning (DL) for cybersecurity applications. AI-based methods have demonstrated strong

capabilities in pattern recognition, anomaly detection, and adaptive learning, making them highly suitable for detecting sophisticated cyber threats [9]. Traditional ML approaches typically require manual feature extraction and feature selection processes, which depend heavily on human expertise. In such systems, feature engineering and classification are often performed as separate stages.

DL models overcome this limitation by automatically learning hierarchical representations of data and performing feature extraction and classification simultaneously, thereby improving detection accuracy and efficiency [10]. Unlike conventional ML models that rely on manually engineered features or third-party feature extraction tools, DL architectures automatically learn discriminative representations directly from raw data [11]. Furthermore, DL models are particularly effective when dealing with large-scale datasets, such as voice recognition, image classification, and phishing detection tasks, as they provide high performance and end-to-end learning capabilities. However, selecting the most appropriate algorithm for a given application remains challenging, and using an unsuitable model may significantly reduce detection accuracy and system efficiency [12].

Phishers continuously evolve their attack strategies by exploiting system vulnerabilities and users' lack of awareness. As a result, early detection of phishing threats has become a critical requirement in modern cybersecurity systems. To address this issue, numerous anti-phishing detection techniques have been developed, and DL-based security solutions are increasingly being adopted across various sectors to combat emerging phishing threats [13]. Researchers have proposed several approaches for detecting phishing websites by leveraging the ability of DL models to extract meaningful features for URL classification. Many detection systems rely on a combination of lexical, structural, and statistical features derived from URLs, such as URL length, keyword frequency, character distribution, and domain-based characteristics.

Major contributions of this work

The main contributions of this research are summarized as follows:

- **Generative encoder with variational generative modeling (VGM):**

A novel generative encoder integrated with a VGM layer is proposed to enforce probabilistic regularization within the latent space. This architecture

enables the model to learn compact and semantically meaningful latent representations directly from raw URL strings.

- **Self-evolving generative threat engine (SE-GTE):**
A novel SE-GTE is introduced to model the dynamic and evolving nature of phishing URLs within the latent space. The SE-GTE learns class-conditional probabilistic representations of benign and phishing URLs and continuously refines its threat knowledge by incorporating both historical and newly observed URL patterns, enabling adaptive threat evolution.
- **Likelihood-based feedback learning strategy:**
A likelihood-based feedback learning mechanism is developed to evaluate both real and synthetically generated latent samples. This strategy selectively reinforces high-confidence phishing patterns, improving the model's ability to detect emerging phishing threats.
- **Hybrid generative–discriminative classifier:**
A hybrid generative–discriminative classifier is designed to jointly leverage discriminative posterior confidence and generative distributional consistency. The classifier incorporates an exponential-family coupled softmax (EF-CS) layer, which unifies generative and discriminative learning within a single probabilistic framework, thereby enhancing classification performance and robustness.

II. Related Works

The Internet has gradually entered all spheres of life in modernity. The current situation of internet-based information and communication technology (ICT) is prone to a myriad of attacks and infiltrations and the loss of a lot of money. In [14], the dual-branch CNN-based multi-modal framework, when trained to extract representations on raw URLs and HTML features and achieves a high classification accuracy of 99.53%, is based on the convolutional feature extraction, which inhibits the model to capture dynamic lexical patterns and long-range sequential dependencies that are common in dynamically generated phishing URLs. In order to overcome this weakness [15], proposes a joint gradient boosting/recurrent neural network ensemble architecture, with gradient boosting and RNNs respectively approximating the static, interpretable attributes of URLs and character-level sequential dependencies, respectively.

Nevertheless, even with better sequential learning, the ensemble quality of the model contributes to complexity in computations and does not have an explicit process of comparison of similarities between benign and phishing URLs at a representational level. In order to address the lack of direct similarity learning [16], suggests a Siamese network to share the weight between two long short-term memory (LSTM) subnetworks to allow robust feature representation learning through direct modeling of the relational distance between classes of URLs. Although this method enhances phishing and legitimate URL separation, it works more with sequence similarity and is not as essential in uncertainty management and rule-based reasoning, which are essential in the real-world noisy phishing dataset. To fill this gap [17], combines Neyman similarity with an Autoencoder with fuzzy neural network stacking and a deep belief network that is more robust due to the use of fuzzy reasoning and uncertainty modeling. However, the use of handcrafted similarity metrics and the use of many stacked operations makes the model more complex and prevents the scaling to large-scale and real-time implementation. In an effort to minimize the complexity of architectural design and increase the robustness of feature learning [18], proposes hybrid spatial-sequential attention CharGRU model which has a logically constrained neural network. Hybrid spatial-sequential attention module is suitable in capturing multi-scale URL invariance and logical constraints enhance the generalization through prior knowledge. Although these have the benefits, the model mostly works at the character-sequence level and does not explicitly utilize contextual bidirectional dependencies, which are important in making sense of phishing URLs containing obfuscated token structures. This is limited in [19], where a BiLSTM-gated highway attention CNN model captures contextual associations on either side, but at the cost of allowing fast convergence and low response time of 12.46 ms. Nonetheless, the deep hybrid architecture adds overhead in terms of memory and might fail to deploy in resource-limited environments. To strike a golden mean between detection efficiency and computation efficiency [20], suggests a hybrid A decision-tree model based on logistic regression, support vector classifier, as well as soft and hard voting. Despite good performance and performance, this assemblage is limited in its ability to automatically learn high-level representations out of the unstructured URL data due to the fact that it uses traditional ML classifiers. In order to enhance the learning of representations [21], proposes models that integrate

both the LSTM and deep neural networks to perform a better task in terms of temporal modeling and non-linear feature extraction. However, such models remain based upon fixed sets of features and are not a complete character-level learning system. In order to remove the need to rely on handcrafted features altogether [22], introduces a new neural architecture where the process of discriminative multi-level character discovery can be performed through the end-to-end concatenation of four-channel information flow through cascading CNN, LSTM, and attention layers. In any case, the design achieves a competitive accuracy of 98.27, it does not explicitly involve adaptive feedback learning and generative pattern generalization which are necessary in detecting zero-day phishing attacks and quickly changing patterns of threats.

III. Proposed Methodology

In the current paper, we present URL-Phish, a phishing URL-detection algorithm based on generative AI and learns distribution-aware latent representations of raw URLs directly. It is an end-to-end pipeline, which consists of: (i) acquisition and annotation of the data, (ii) pre-process the URLs and standard forms, (iii) learning features of a variational latent model, and (iv) phishing classification with hybrid generative-discrimination decision module, which is complemented by a self-evolving threat engine. The aim is to improve generalization to both the zero-day phishing URLs and adversarial phishing URLs, by training a continuous latent manifold, which has been learned to capture structural variation. The proposed VGM framework is conceptually inspired by variational autoencoders (VAEs) but differs in both formulation and objective. Unlike standard VAEs, which assume a fixed isotropic Gaussian prior $p(z) = N(0, I)$ and focus on reconstructing input data, VGM employs class-conditional latent distributions $p(z | y = c) = N(\mu_c, \Sigma_c)$ to explicitly model the probabilistic structure of benign and phishing URL representations. Furthermore, VGM is not designed for data reconstruction; instead, it learns discriminative latent embeddings that facilitate probabilistic threat scoring and synthetic threat generation. This enables seamless integration with the SE-GTE, allowing the model to dynamically adapt to evolving phishing patterns. Therefore, VGM can be interpreted as a task-specific extension of VAE principles tailored for classification and adversarial threat modeling rather than a direct implementation of a conventional VAE.

a. Dataset description

The URL-Phish dataset that the current research will be based is URL-Phish dataset [23], which is comprised of 111,660 URLs (11,660 phishing [label = 1] and 100,000 non-phishing [label = 0]). This was compiled in a systematic process that involved information collection, pre-processing, generation of representation and data segmentation, and validation of the baseline.

- **Benign subset (100,000 samples):** Benign URLs were found on popular domains, such as.edu,.gov and Alexa Top Sites. The subset was downloaded to a publicly available Zenodo repository of DOI-registered URL listings and provides reliability and validity.
- **Phishing subclass (11,660 samples):** Phishing URL sources PhishTank is a threat intelligence community site where phishing URLs reports are assembled and confirmed. Phishing category includes the date between November 12, 2024 and September 24, 2025, thus it is important to ensure that there is time gap between phishing campaign and attacks.

b. Preprocessing of dataset

The dataset undergoes preprocessing through data scraping, integration, duplicate removal, and character-level tokenization to preserve the structural integrity of URLs. Each raw URL string v is decomposed into an ordered sequence of characters:

$$v = [c_1, c_2, \dots, c_n]$$

To explicitly capture sequence boundaries, special tokens are appended, resulting in:

$$[\langle \text{STA} \rangle, c_1, c_2, \dots, c_n, \langle \text{SEP} \rangle]$$

To maintain uniform input dimensionality, each sequence is normalized to a fixed length d . Sequences longer than d are truncated, while shorter sequences are padded accordingly, yielding an indexed representation:

$$x \in \mathbb{R}^d$$

Subsequently, the indexed tokens are transformed into dense vector embeddings of dimension h , resulting in a fixed-size URL representation:

$$V \in \mathbb{R}^{d \times h}$$

This representation preserves sequential and structural characteristics of the URL while ensuring consistency across all samples, thereby serving as input to the subsequent generative feature extraction and classification modules.

Finally, dense representations of dimension h are indexed and made consistent across the samples, and they act as inputs to the subsequent generative feature extraction and classification modules.

Representation learning is also carried out over the preprocessing in order to perform the feature analysis and selection rather than discarding the features manually. Character level embedding preserves the signs of the lexical and structure, but truncation and padding fixed length reduce the noise of information introduced by the difference in length of URLs. The dimensionality reduction is further implemented in the generative encoder through projection and residual blocks which are the dimensionality reduction of the representation of URLs in the high dimensions into small embeddings without alteration of the discriminative patterns. These embeddings are then converted to a low-dimensional probabilistic latent space with the assistance of the VGM that lowers the number of unneeded variations without harming the structures connected to phishing. The hierarchical compression (raw URL—embedding—latent vector) is associated with reduced computation cost and time, model efficiency, and high resistance to the shifting phishing trends.

c. Generative AI-driven representation learning

In the study, generative AI is employed to obtain meaningful trends on the raw URL data by modeling underlying structure of phishing behavior rather than just modeling the limits of classification. However, unlike the conventional approaches, the generative learning assists the system to comprehend alteration and obscurity in the design of URLs and this provides the system with greater level of adaptability to the more pliable phishing plans. It is aimed at the conversion of uncoded URLs into informational latent forms with structural relationships.

$$Z = G(X)$$

that is, X is the input URL and Z is the learned latent representation.

The URL representation is coded using a help of a generative encoder as illustrated below.

c.i. Generative encoder for URL representation

Generative encoder can extract both structural and sequential features in the URLs. It does not acquire lexical dependencies, patterns of obfuscation and implicit character relationships through engineered aspects. This allows the model to form small representations with difference between benign and phishing URLs.

$$L = L_{gen} + \lambda L_{cls} \quad E = f(X)$$

X represents the input URL, whereas E denotes the encoded feature representation.

Algorithm 1: Generative Encoder for URL Representation

Input: Embedded URL tensor $V^{(i)} \in \mathbb{R}^{d \times h}$, optional auxiliary cues $\{a_t^{(i)}\}_{t=1}^T$, static cues $static^{(i)}$

Output: Deterministic embedding $comp_emb^{(i)}$
Steps:

1. **Feature projection:** For each $t = 1..T$, project auxiliary cues using residual blocks:
 2. $a_t^{(i)} = residual_block(a_t^{(i)})$ with $l' \ll l$.
 3. Stack projected cues $a'_{1:T}^{(i)}$.
 4. Concatenate encoder inputs:
 5. $X^{(i)} \leftarrow Concat(V^{(i)}; a'_{1:T}^{(i)}; static^{(i)})$
 6. Pass through deep residual encoder with depth $hyper_e$:
 7. $comp_{emb}^{(i)} \leftarrow (X^{(i)})$
 8. Return $comp_{emb}^{(i)}$.
-

c.ii. Variational generative modeling for pattern generalization

Variational modeling can help improve the performance of the representation by using uncertainty learning. Other models have the limitation of using the features learned as constants. However, in variational modeling, the use of probabilistic patterns helps in the

representation of the features learned. This helps in the generalization of unseen phishing attacks and the robustness of the model against zero-day attacks.

$$Z = \mu + \sigma \cdot \epsilon$$

with σ being the mean and μ being the variance, and ϵ is random noise.

Algorithm 2: Variational Generative Modeling (VGM)

Input: Deterministic embedding $comp_{emb}^{(i)}$.

Output: Latent sample $z^{(i)}$, posterior parameters $\mu^{(i)}, \sigma^{(i)}$

1. Receive deterministic embedding $comp_{emb}^{(i)}$ from the generative encoder.
2. Generate the latent posterior distribution conditioned on $comp_{emb}^{(i)}$ using Eq. (3):
3. Convert Extract $\mu^{(i)}$ and $\sigma^{(i)}$ from the variational encoder network (mean and variance vectors):
4. Sample Gaussian noise $\epsilon \sim N(0, I)$ (Eq. (4))
5. Obtain the latent sample using the reparameterization trick in Eq. (4):
6. Decode $z^{(i)}$ to reconstruct the URL representation (reconstruction is referenced by Eq. (5) through $\log p_{\theta}(x | z)$)
7. Compute the ELBO training objective using Eq. (5):

$$Loss_{ELBO} = E_{q_{\phi}(z|x)} [\log p_{\theta}(x|z)] - Kull(q_{\phi}(z|x) || p(z))$$

8. Compute Optimize model parameters by maximizing ELBO (or minimizing $-Loss_{ELBO}$) to encourage reconstruction + latent regularization.
 9. Return $z^{(i)}, \mu^{(i)}, \sigma^{(i)}$ and $Loss_{ELBO}$
-

d. Generative AI classification

Following the extraction of distribution-aware latent features using the generative encoder and variational modeling, the proposed system executes phishing detection through a classification module driven by generative AI. This module is engineered to maintain efficacy against advancing phishing tactics through the integration of a SE-GTE and a hybrid

generative–discriminative classifier, as depicted in Figure 1.

d.i. SE-GTE design

The SE-GTE continuously learns phishing behavior and generates synthetic threat patterns in the latent space. This helps the system anticipate new phishing strategies and enhances adaptability by expanding the learning boundary with simulated attack patterns.

$$T = Z + \delta$$

where Z is the latent representation and δ represents controlled perturbation used to generate new threats.

Algorithm 3: Self-Evolving Generative Threat Engine (SE-GTE)

Input: URL sample u_i , latent dimension d , evolution noise scale τ , confidence threshold δ

Output: High-confidence synthetic threats $\{z_j'\}$ and threat scores $S(\cdot)$

1. Receive input URL u_i from the pipeline.
2. Map the URL to latent embedding using Eq. (13):
3. Model benign and phishing latent behavior using Eq. (14):
4. Sample synthetic phishing candidates from the phishing distribution
5. Apply controlled perturbation to diversify synthetic threats (Eq. (15)):
6. Compute the probabilistic threat confidence score using Eq. (16):

$$z_i = E_{\phi}(u_i), z_i \in R^d$$

$$p(z|y=c) = N(\mu_c, \sum c), \text{ where } c \in \{0, 1\}$$

$$z_j' \sim N(\mu_1, \sum 1),$$

$$z_j' = z_j' + \epsilon, \epsilon \sim N(0, \tau I)$$

$$S(z) = \log p(z|y=1) - \log p(z|y=0)$$

7. Select only high-confidence synthetic threats using Eq. (17):

$$S(z_j') \geq \delta$$

8. Add selected z_j' into the evolving threat pool to enrich the classifier decision boundary.
 9. Output the high-confidence threat embeddings and corresponding $S(\cdot)$ scores.
-

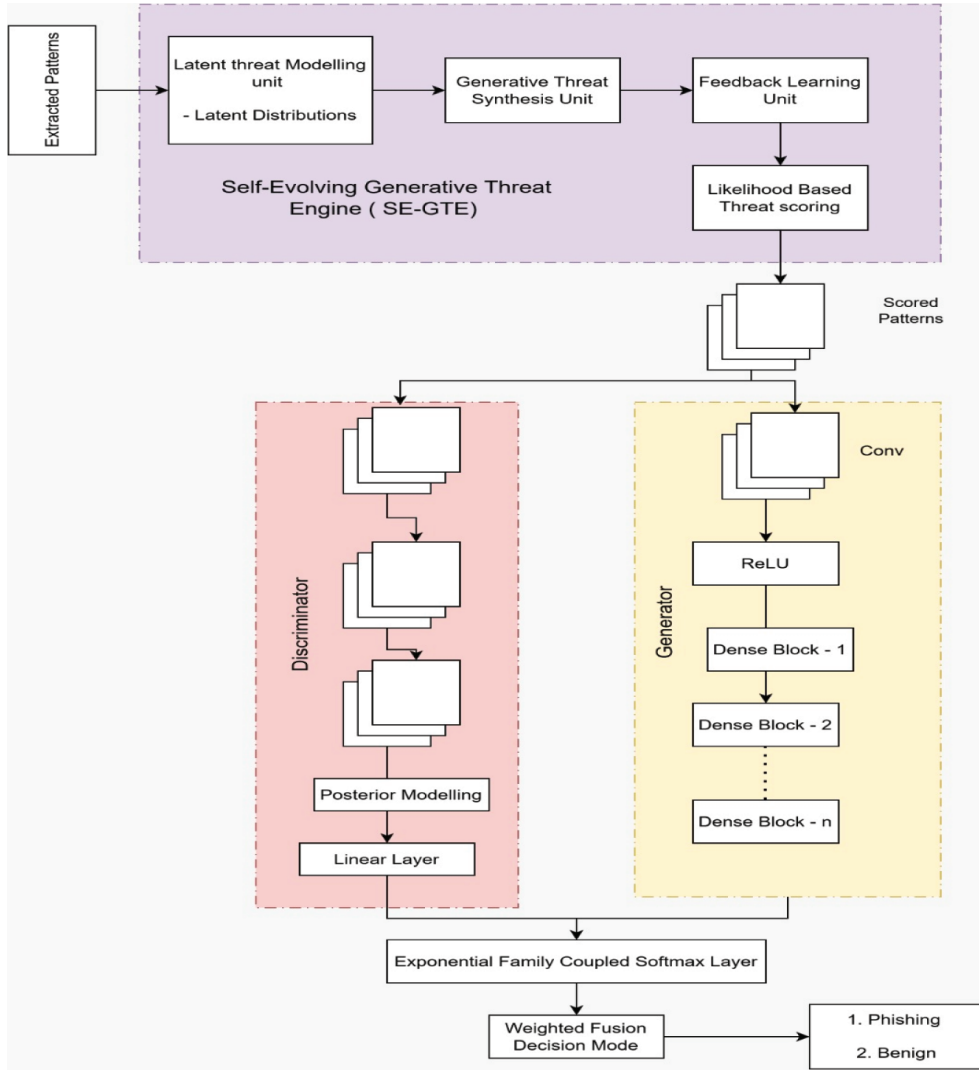


Figure 1: Architecture of hybrid generative-discriminative classifier with SE-GTE. SE-GTE, self-evolving generative threat engine.

d.ii. Hybrid generative-discriminative classifier

This classifier combines the knowledge of patterns from the generative learning paradigm and the ability to predict from the discriminative learning paradigm. It makes a stable and accurate classification decision despite the changing patterns of phishing.

$$P(y|Z) = \alpha P_d + (1 - \alpha) P_g$$

where P_d is discriminative probability and P_g is generative likelihood.

Algorithm 4: Hybrid Generative-Discriminative Classifier (EF-CS + Fusion)

Input: Latent feature $y = z$, fusion weight α

Output: Predicted class label $\hat{c} \in \{0,1\}$

1. Receive latent representation $y \equiv z$ from the VGM output.
2. Compute discriminative class probability using the EF-CS softmax formulation in Eq. (23):

$$p(c|y, \Delta) = \frac{\exp(w_c^T y + b_c)}{\sum_{c'=1}^C \exp(w_{c'}^T y + b_{c'})}$$

-
3. Compute generative modeling signal through exponential-family likelihood in Eq. (24):

$$p(y, c | \Delta_c) = \pi_c h(y) \exp(\eta_c^T T(y) - A(\eta_c))$$

4. Use both discriminative and generative signals for decision making.
5. Fuse both signals using the weighted decision criterion given in Eq. (25):

$$\hat{c} = \arg \max_c [\alpha \log p(c | y) + (1 - \alpha) \log p(y | c)]$$

6. Evaluate the fused score for $c = 0$ (benign) and $c = 1$ (phishing).
 7. Select the class $\hat{c}$ that maximizes the fused criterion.
 8. Output the predicted label $\hat{c}$ along with confidence.
 9. Return $\hat{c}$.
-

e. Model building

The paper introduces a hybrid phishing URL recognition model, which is trained on a generative AI that integrates representation learning, probabilistic generalization, adaptive threat synthesis, and confidence-sensitive categorization into a single model. The last recommended one is an integrated version of four linked modules, that is: (i) Generative Encoder to encode URL, (ii) VGM, (iii) SE-GTE, and (iv) hybrid generative-discriminative classifier (EF-CS + Fusion). The combination of these modules is a workflow that learns resilient latent patterns on unprocessed URLs and is robust to changing the phishing policy and a zero-day phishing approach.

e.i. Hybrid architecture and workflow of proposed model

This begins with preprocessing URLs, encoding, probabilistic modeling, threat synthesis and classification. This kind of multi-level system improves the security of the detection and allows the system to handle new phishing attacks.

$$Y = f(Z)$$

where component predictive label Y predicted and learned latent feature Z .

f. Training setup and experimental protocol

The stratified data splits are utilized to train the model so as to secure the balance of classes and improve the quality of learning. The challenge is on maximizing the output of class imbalanced detectors and adapting to evolving phishing activities.

$$L = L_{gen} + \lambda L_{cls}$$

where L_{gen} is generative loss and λL_{cls} is classification loss.

The stratified data splits are utilized to train the model so as to secure the balance of classes and improve the quality of learning. The challenge is on maximizing the output of class imbalanced detectors and adapting to evolving phishing activities.

g. Training configuration and hyperparameters

The model was trained using the Adam optimizer with an initial learning rate of 0.001 and a batch size of 64. The training process was conducted for 50 epochs with early stopping based on validation loss to prevent overfitting. The latent representation dimension was set to 128, and dropout regularization with a rate of 0.3 was applied in the encoder layers. The overall training objective combines the variational generative loss and classification loss using a balancing parameter λ to ensure stable joint optimization of generative representation learning and discriminative classification.

The model was trained using the Adam optimizer with an initial learning rate of 0.001 and a batch size of 64. Training was conducted for 50 epochs, with early stopping applied based on validation loss to mitigate overfitting. The latent representation dimension in the generative encoder was set to 128 to capture meaningful feature abstractions from the input data. Dropout regularization with a rate of 0.3 was incorporated within the encoder layers to improve generalization and reduce the risk of overfitting. The dataset was partitioned using an 80:10:10 split for training, validation, and testing, respectively, while maintaining class balance through stratified sampling. The overall objective function jointly optimizes the generative reconstruction loss, the Kullback–Leibler divergence regularization, and the classification loss, balanced through a weighting parameter λ to ensure stable learning between generative representation modeling and discriminative prediction.

Algorithm 5: VGA-HyPhishNet: Variational Generative Hybrid Model for Phishing URL Detection

Require: Raw URL dataset $D = \{(u_i, c_i)\}_{i=1}^N$, latent dimension d , fusion weight α , loss trade-off λ

Ensure: Predicted class label $\hat{c}_i \in \{0, 1\}$

- 1: **Preprocessing:**
- 2: for each URL $u_i \in D$ do
- 3: Tokenize URL into character sequence
- 4: Encode characters into embedding tensor
- 5: Extract auxiliary URL features (lexical, structural, and statistical)
- 6: end for
- 7: **Generative URL Encoding:**
- 8: for each encoded URL do
- 9: Project auxiliary features using residual projection blocks
- 10: Concatenate character embeddings and projected features
- 11: Generate compact embedding e_i using generative encoder
- 12: end for
- 13: **Variational Generative Modeling:**
- 14: for each compact embedding e_i do
- 15: Infer variational posterior:

$$q_{\phi}(z_i | e_i) = N(\mu_i, \sigma_i^2 I)$$
- 16: Sample latent representation via reparameterization:

$$z^{(i)} = \mu^{(i)} + \sigma^{(i)} \odot \epsilon, \epsilon \sim N(0, 1)$$
- 17: end for
- 18: Optimize variational generative objective using:

$$Loss_{ELBO} = E_{q_{\phi}(z|x)} [\log p_{\theta}(x|z)] - Kull(q_{\phi}(z|x) || p(z))$$

- 19: **Self-Evolving Generative Threat Engine (SE-GTE):**
- 20: Estimate class-conditional latent distributions for benign and phishing URLs
- 21: Generate synthetic phishing embeddings via controlled perturbation
- 22: Select high-confidence synthetic threats based on threat scoring
- 23: **Hybrid Generative-Discriminative Classification:**
- 24: for each latent representation z_i do
- 25: Compute discriminative posterior score $p(c|z_i)$
- 26: Compute generative likelihood score $p(z_i | c)$
- 27: Fuse decisions using:

$$\hat{c} = \arg \max_c [\alpha \log p(c|y) + (1 - \alpha) \log p(y|c)]$$

- 28: end for
- 29: **Model Optimization:**
- 30: Minimize total objective:

$$L_{total} = L_{ELBO} + \lambda L_{cls}$$

- 31: Apply class-weighted classification loss to address imbalance
 - 32: **return** Predicted labels for all test URLs
-

h. Optimization strategy

The proposed VGA-HyPhishNet model is trained using a composite loss function that integrates generative and discriminative objectives. The overall loss is defined as:

$$L_{total} = \alpha L_{rec} + \beta L_{KL} + \gamma L_{cls}$$

where L_{rec} denotes the reconstruction loss, L_{KL} represents the Kullback–Leibler divergence enforcing latent regularization, and L_{cls} corresponds to the classification loss (e.g., binary cross-entropy). The coefficients α , β , and γ are hyperparameters that control the contribution of each component.

The reconstruction loss ensures that the latent representation preserves essential structural information from the input URL sequences, while the KL-divergence term regularizes the latent space to follow the desired probabilistic distribution. The classification loss drives the model toward discriminative performance for phishing detection.

In practice, the weights are empirically tuned to balance generative fidelity and classification accuracy. A moderate weighting of the KL-divergence term is used to avoid posterior collapse, while a relatively higher emphasis is placed on the classification loss to ensure robust phishing detection performance. All models are trained using stochastic gradient-based optimization with consistent hyperparameter settings across experiments.

i. Model evaluation

The evaluation of model performance employs both comprehensive and class-specific metrics. We present the four metrics of accuracy, precision, recall, and F1-score; also, the weighted metrics of recall, accuracy, and precision are employed to address class imbalance by averaging per-class metrics based on their support (the quantity of samples for each category). The confusion matrix has four possible outcomes: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

Accuracy: Accuracy measures the proportion of correctly classified samples over the entire test set:

$$\text{Accuracy} = \frac{TP + TN}{S} \quad (26)$$

Precision: quantifies the accuracy of positive forecasts by calculating the percentage of phishing URLs that are truly phishing (penalizes FP):

$$\text{Precision} = \frac{TP}{TP + FP} \quad (27)$$

Recall: A classifier’s recall, sensitivity, or true positive rate (TPR) indicates how well it can identify all actual phishing URLs (penalizes FN):

$$\text{Recall} = \frac{TP}{TP + FN} \quad (28)$$

F1-Score: In order to evaluate a classifier fairly, the F1-score takes the harmonic mean of the two metrics, precision and recall. It really shines when working with datasets that aren’t balanced, where precision or recall alone may be insufficient to represent model performance.

$$F_1 - \text{Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (29)$$

IV. Results and Discussion

A thorough assessment and examination of the canonical proposed VGA-HyPhishNet in relation to the existing methods in the detection of URL-based phishing attacks. The results are analyzed in terms of various performance metrics, confusion matrices, ROC curves, computational costs, etc., in terms of the accuracy, robustness against class imbalance, and efficiency of the detection mechanism. The discussion also sheds light on the implications of the proposed framework in terms of the advantages, trade-offs, and consequences of the proposed hybrid mechanism in detecting phishing attacks. To ensure statistical robustness and reduce bias due to data partitioning, k-fold cross-validation was employed. Specifically, the dataset was partitioned into k equal subsets, and the model was trained and evaluated k times, with each subset used once as the test set while the remaining $k - 1$ subsets were used for training. The reported performance metrics correspond to the average across all folds, providing a reliable and generalized estimate of model performance.

For fair evaluation, all baseline models (URL-Transformer, BiLSTM-URL, and CharCNN-URL) were

re-implemented under identical experimental conditions as in Table 1, including the same preprocessing pipeline, dataset splits, and evaluation metrics.

Table 2 presents the comparative performance of the proposed VGA-HyPhishNet against widely used URL-based phishing detection models. The proposed model achieves superior performance across all evaluation metrics, with an accuracy of 0.9873, precision of 0.9300, recall of 0.9497, and an F1-score of 0.9397. These results indicate its strong capability in accurately detecting phishing URLs while minimizing both FP and FN.

In comparison, the URL-Transformer attains an F1-score of 0.7899, reflecting limited robustness in handling class imbalance and evolving phishing patterns despite its ability to capture global token dependencies through self-attention. The BiLSTM-URL model demonstrates moderate recall but suffers from lower precision, resulting in an overall F1-score of 0.7477, indicating challenges in discriminating subtle phishing characteristics. Similarly, the CharCNN-URL model records the lowest F1-score of 0.7267, primarily due to its limited capacity to capture long-range dependencies in URL sequences.

The confusion matrices as in Figure 2 exhibit the classification behavior of the proposed VGA-HyPhishNet and the baseline models on benign and phishing URLs. With 14,875 benign and 1,661 phishing URLs accurately identified and the fewest FP (125) and FN (88), VGA-HyPhishNet exhibits the most balanced and dependable performance, demonstrating strong robustness against both forms of misclassification. The URL-Transformer, on the other hand, shows a greater number of FP (394) and FN (350), indicating decreased stability under changing phishing trends. The BiLSTM-URL and CharCNN-URL models exhibit significant degradation, with substantially larger false-negative counts (368 and 385, respectively), indicating their inadequate capacity to consistently detect phishing URLs. Overall, the confusion matrix analysis shows that, as compared to traditional discriminative methods, the generative representation learning, adaptive threat synthesis, and hybrid decision fusion used in VGA-HyPhishNet greatly minimize misclassification errors.

The ROC curves presented in Figure 3 illustrate the trade-off between the TPR and false positive rate (FPR) for the proposed VGA-HyPhishNet model and the baseline approaches across different decision thresholds. From the figure, it is evident that the proposed VGA-HyPhishNet model outperforms all other models, achieving the highest AUC value of

Table 1: Example transformation of a URL through the proposed feature extraction stages

Stage	Sample output
Raw URL	'https[//]www[.]amazon[.]com/gp/help/customer/display.html'
Character Tokenization	<STA> h t p s [/ /] w [.] a m a z o n ... h t m l <SEP>
Character Encoding	[2,11,23,23,19,22,42,40,40,43,27,27,27,...,28,3]
Generative Encoder Output	[0.41, -0.27, 1.13, 0.08, -0.66, ..., 0.09]
Variational Encoder (Mean)	[0.38, -0.31, 1.05, 0.12, -0.60, ..., 0.11]
Variational Encoder (Variance)	[0.04, 0.06, 0.02, 0.05, ..., 0.04]

Table 2: Comparative performance of the proposed VGA-HyPhishNet and baseline models

Model	Accuracy	Precision	Recall	F1-score
VGA-HyPhishNet	0.9873	0.9300	0.9497	0.9397
URL-Transformer	0.9556	0.7803	0.7999	0.7899
BiLSTM-URL	0.9444	0.7100	0.7896	0.7477
CharCNN-URL	0.9387	0.6803	0.7799	0.7267

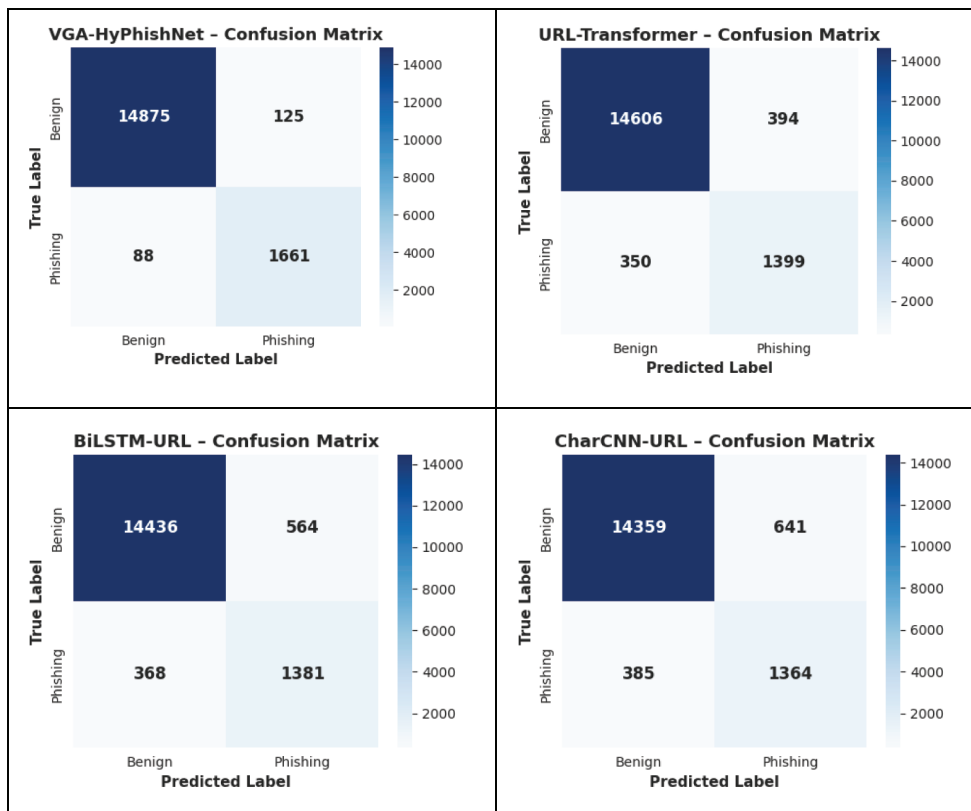


Figure 2: Confusion matrix.

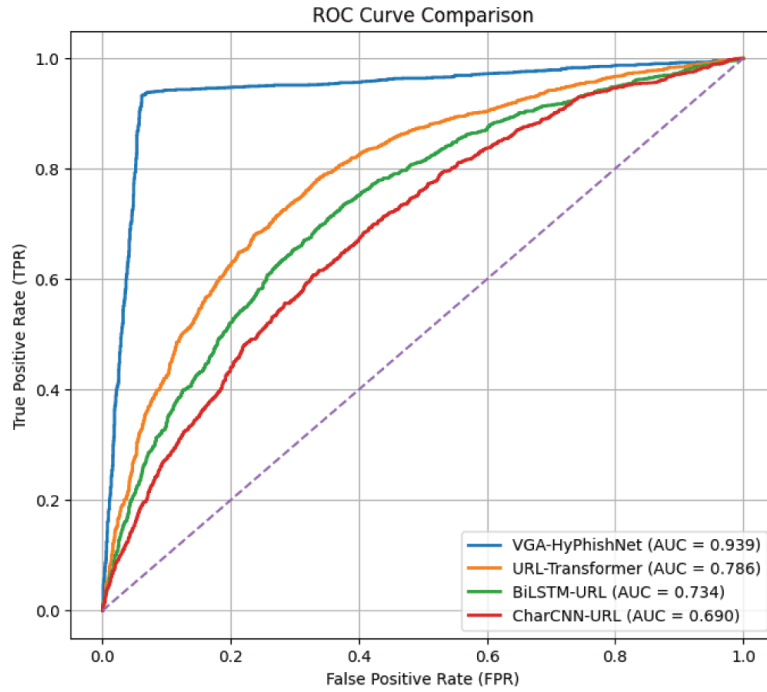


Figure 3: ROC curve. FPR, false positive rate.

0.939. This indicates strong discriminative capability and robustness in distinguishing phishing URLs from legitimate ones, even at relatively low FPRs.

In comparison, the URL-Transformer model demonstrates moderate performance, with an AUC of 0.786, reflecting its limited ability to capture subtle variations in phishing URL patterns. Furthermore, the BiLSTM-URL and CharCNN-URL models exhibit comparatively lower performance, with AUC values of 0.734 and 0.690, respectively. These lower values suggest reduced generalization capability and weaker sensitivity to evolving and obfuscated phishing attacks.

Figure 4 shows the relative suitability of each model for real-time deployment based on the average inference time per URL. From the results, the CharCNN-URL model has the lowest inference time of approximately 15 ms. The reason for this is the simple convolutional processing, where the computational cost is linearly increasing with the size of the URL. The second lowest inference time is obtained by the VGA-HyPhishNet model with an average inference time of approximately 18 ms. This indicates that the costs of using the generative encoding, variational sampling, and hybrid fusing are marginal. The highest inference time is obtained by the BiLSTM-URL model with an average inference time of approximately 35 ms. The reason for this is the sequential

processing of the characters in the forward and backward directions. The URL-Transformer model has the highest inference time of approximately 55 ms, showing the cost of using self-attention mechanisms, which have quadratic computational costs in the size of the input sequence.

Using a dataset of roughly 100,000 URLs, this Figure 5 shows the training time per epoch for various phishing detection algorithms, illustrating their respective computational cost during learning. CharCNN-URL shows the lowest training cost at roughly 380 s per epoch, owing to its shallow convolutional architecture and excellent parallelization. VGA-HyPhishNet requires a slightly greater training time of roughly 400 s per epoch, which is ascribed to the added overhead of variational latent modeling and hybrid generative–discriminative learning, yet it remains computationally viable. BiLSTM-URL has a much higher training cost of around 1,300 s per epoch due to sequential processing and bidirectional recurrence over URL characters. Due to its high parameter counts and self-attention actions, URL-Transformer has the highest training cost at almost 2,500 s each epoch. These results reveal that VGA-HyPhishNet delivers good detection performance with a training cost close to lightweight CNN models, while being substantially more efficient than recurrent and transformer-based alternatives.

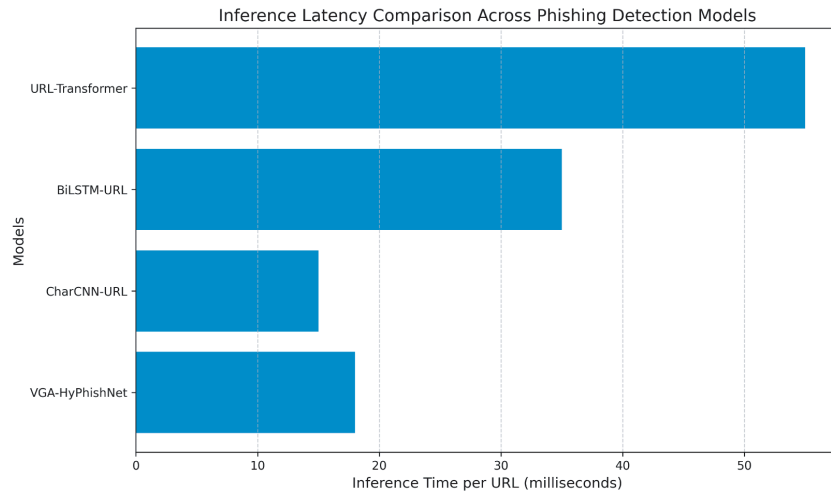


Figure 4: Inference time comparison across phishing detection models.

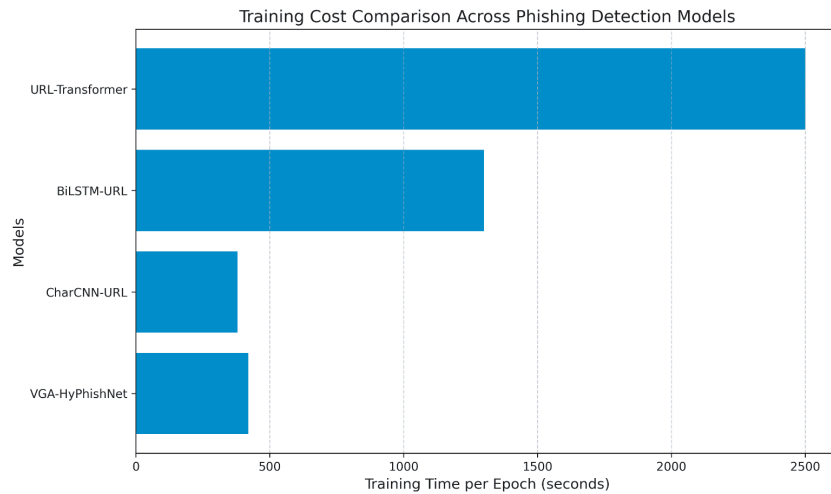


Figure 5: Training cost comparison across phishing detection models.

The ablation research that was done to assess each component's contribution to the suggested VGA-HyPhishNet system is shown in Table 3. The results show that overall, the model is the most effective overall, highlighting the significance of integrating all parts. The model performs worse when variational modeling is eliminated because it is less able to generalize to hidden phishing patterns. In a similar vein, the system's capacity to learn from artificial phishing patterns is diminished when the SE-GTE is removed. Generative-only modeling captures structural behavior but lacks a strong decision-making capacity, whereas discriminative classification alone reduces accuracy by increasing FP. Although it works quicker, the discriminative-only variation is less stable

when phishing patterns change. Overall, the research demonstrates that detection accuracy and robustness are greatly increased by the hybrid combination of generative learning, adaptive threat modeling, and categorization.

a. Model interpretability and decision analysis

To provide insights into the model's decision-making process, an analysis of the learned latent representations and input-level patterns was conducted. At the latent level, the VGM module learns class-conditional distributions that capture structural differences between benign and phishing URLs. Phishing URLs

Table 3: Ablation study and parameter sensitivity of proposed study

Setting/variant	Key change	Accuracy	Precision	Recall	F1-score
Full Model (VGA-HyPhishNet)	Encoder + Variational Modeling + SE-GTE + Hybrid Fusion	0.9873	0.9300	0.9497	0.9397
w/o VGM (No variational modeling)	Variational modeling removed; deterministic features used	0.9728	0.8850	0.9040	0.8944
w/o SE-GTE	Synthetic threat generation removed	0.9762	0.9021	0.9215	0.9117
w/o Hybrid Fusion	Only discriminative classification used	0.9709	0.8724	0.9280	0.8993
Generative-only	Only generative decision used	0.9584	0.8202	0.9020	0.8591
Discriminative-only	Only classification-based decision used	0.9687	0.8658	0.9149	0.8897

SE-GTE, self-evolving generative threat engine.

tend to occupy regions of the latent space characterized by higher likelihood under $p(z|y=1)$, enabling separation through log-likelihood ratio-based scoring.

At the input level, the model implicitly captures discriminative URL characteristics through character-level embeddings. Key indicative patterns include the presence of suspicious tokens (e.g., excessive special characters, uncommon domain structures), abnormal sequence lengths, use of obfuscated strings, and irregular subdomain compositions. These features contribute to distinctive latent embeddings, which are further amplified by the SE-GTE through synthetic threat augmentation.

Additionally, gradient-based attribution analysis (e.g., saliency mapping) indicates that specific character sequences, such as encoded symbols, repeated delimiters, and uncommon domain suffixes have a higher contribution to phishing classification. This demonstrates that the model effectively leverages both structural and probabilistic patterns for robust decision making.

b. Discussion

The results of the experiment demonstrate the efficacy of the proposed VGA-HyPhishNet in enhancing the phishing URL detection rate with the help of a hybrid generative-discriminative learning algorithm. Having high results on all testing scores compared to the conventional models, such as URL-Transformer, BiLSTM-URL, and CharCNN-URL, where VGA-HyPhishNet has a high capacity to distinguish phishing URLs and authentic URLs,

and reduction of the false positive and negative. The confusion matrix presents the further evidence of the stability analysis of the model that shows that the misclassification rates have reduced to a much lower degree and that the stability of the model to the altering phishing patterns and class imbalance is considerable. VGA-HyPhishNet implements VGM to learn the hidden structural representation of URLs by enabling more generalization in the face of unknown threats as compared to standard discriminative designs that simply rely on the observable data patterns. VGA-HyPhishNet possessed the highest AUC, meaning that there is improved separability of phishing and benign classes even with high decision thresholds which is supported by the ROC analysis. Computationally, this model is still practically efficient, and the speed of inference is comparable to lightweight CNN-based models, but significantly faster than transformer and recurrent models. Although the increment in generative elements may increase the cost of training by a small margin, the increase in detection accuracy and freedom justifies the trade-off. The ablation research also contributes to the importance of every architectural component by demonstrating that the removal of variational modeling, synthetic threat generation, or hybrid fusion would lead to the visible decline in performance. The hybrid decision process is particularly significant to create a balance between accuracy and recall, in order to offer the constant detection in dynamic phishing situations. Altogether, this discussion indicates that not only does the discriminative classification and generative

representation learning produce a higher detection accuracy, but also enhances against online threats changes, which is why VGA-HyPhishNet is a reliable and scalable choice in relation to real-time phishing detection systems.

V. Conclusion

In this study, a novel hybrid generative-discriminative system was introduced under the name of VGA-HyPhishNet which aims at enhancing the phishing URLs in dynamic and real time settings. The proposed model uses generative representation learning compared to the traditional detection models which simply use the fixed classification boundaries; it tries to learn the aspects of phishing as a probabilistic activity. The system includes A SE-GTE, a generative encoder, VGM, and a hybrid decision classifier, which in the framework of experimental performance outperform current baseline models, in terms of F1-score, recall, accuracy, and precision and maintains an efficient inference time, suitable to be deployed in real-time. The model can anticipate concealed phishing habits by developing the synthetic threats with the help of SE-GTE, which increases its vulnerability to zero-day attacks significantly. Also, the discriminative confidence combined with generative likelihood hybrid combination ensures balanced decision making by reducing FN and FP, as well as enhance classification stability, threat evolution, and variational modeling that boosts the generalization of the system. The ablation study ensured that all architectural components play a crucial role in the work of the system through hybrid fusion, which enhances stability in classification, threat evolution, and variational modeling. When all is said and done, the proposed architecture demonstrates that discriminative learning combined with generative intelligence is a powerful technique toward preventing rapidly evolving phishing attacks.

References

[1] A. K. Jain and B. B. Gupta, "Phishing detection: analysis of visual similarity based approaches," *Secur. Commun. Networks*, vol. 2017, no. 1, p. 5421046, 2017.

[2] S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A framework for detection and measurement of phishing

attacks," in *Proceedings of the 2007 ACM workshop on Recurring malware*, 2007, pp. 1–8.

[3] R. Verma and A. Das, "What's in a url: Fast feature extraction and malicious url detection," in *Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics*, 2017, pp. 55–63.

[4] J. Hong, "The state of phishing attacks," *Commun. ACM*, vol. 55, no. 1, pp. 74–81, 2012.

[5] M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, "Intelligent phishing detection system for e-banking using fuzzy data mining," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 7913–7921, 2010.

[6] K. Kaur and A. K. Jain, "A Survey on Phishing Attack Taxonomy, Detection Techniques, Datasets, and Security Measures," *J. Appl. Secur. Res.*, vol. 20, no. 4, pp. 506–557, 2025.

[7] S. Alarifi and D. Aljeaid, "Cross-Sectional Analysis to Discover Phishing Cyberattack Trends in Saudi Arabia," *J. Ecohumanism*, vol. 4, no. 1, pp. 3830–3843, 2025.

[8] J. Pathmanaban, P. G. James, P. Ashok, B. Ragesh, S. Aakash, and N. Kaushik, "Phishing Website Detection Using Machine Learning," in *2024 2nd International Conference on Networking and Communications (ICNWC)*, IEEE, 2024, pp. 1–5.

[9] J. Heaton, "Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618," *Genet. Program. evolvable Mach.*, vol. 19, no. 1, pp. 305–307, 2018.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," *Knowledge-Based Syst.*, vol. 189, p. 105124, 2020.

[12] F. Thabtah and D. Peebles, "A new machine learning model based on induction of rules for autism detection," *Health Informatics J.*, vol. 26, no. 1, pp. 264–286, 2020.

[13] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE access*, vol. 7, pp. 41525–41550, 2019.

[14] M. Alsaedi, F. A. Ghaleb, F. Saeed, J. Ahmad, and M. Alasli, "Multi-modal features representation-based convolutional neural network model for malicious website detection," *IEEE Access*, vol. 12, pp. 7271–7284, 2023.

- [15] L. Hung, “Urlnet: learning a URL representation with deep learning for malicious URL detection,” *ArXiv*, vol. 1802, p. v2, 2017.
- [16] K. Sruthi, “A novel framework for effective phishing URL detection using an LSTM-based Siamese network,” *Knowledge-Based Syst.*, p. 114271, 2025.
- [17] H. Y. A. Abutair and A. Belghith, “Using case-based reasoning for phishing detection,” *Procedia Comput. Sci.*, vol. 109, pp. 281–288, 2017.
- [18] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J.-P. Niyigena, “An effective phishing detection model based on character level convolutional neural network from URL,” *Electronics*, vol. 9, no. 9, p. 1514, 2020.
- [19] M. Tayebi and S. El Kafhali, “Attention-enhanced BiLSTM-ANN framework with CNN-based feature selection for advanced threat detection,” *Int. J. Mach. Learn. Cybern.*, vol. 17, no. 2, p. 52, 2026.
- [20] 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.
- [21] R. Vinayakumar, K. P. Soman, and P. Poornachandran, “Evaluating deep learning approaches to characterize and classify malicious URL’s,” *J. Intell. Fuzzy Syst.*, vol. 34, no. 3, pp. 1333–1343, 2018.
- [22] G. Li et al., “Audio-visual end-to-end multi-channel speech separation, dereverberation and recognition,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 2707–2723, 2023.
- [23] D. M. Linh and T. C. Hung, “A feature-engineered dataset of benign and phishing URLs for machine learning and large language models evaluation,” *Data Brief*, 2025, doi: 10.1016/j.dib.2025.112162.