Enhanced Phishing URL Detection Using a Novel GRU-CNN Hybrid Approach

[1]Sangeetha M, [2]Navaz K, [3]Santosh Kumar Ravva, [4]Roopa R, [5]Penubaka Balaji and [6]Ravi Kumar T
[1]Department of Information Technology, SRM Madurai College for Engineering and Technology, Sivagangai, Tamil Nadu, India.
[2]Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, Tamil Nadu, India.
[3]Department of Computer Science and Engineering, Vasavi College of Engineering, Hyderabad, Telangana, India.
[4]Department of Computer Science and Engineering (Data Science), Madanapalle Institute of Technology and Science, Madanapalle, Andhra Pradesh, India.
[5]Department of Computer Science and Engineering, K L University, Vaddeswaram, Guntur, Andhra Pradesh, India.
[6]Department of Computer Science and Engineering, Aditya Institute of Technology and Management, Tekkali, Andhra Pradesh, India.
[1]sangeetha.srm.23@gmail.com, [2]navazit@gmail.com, [3]santosh@staff.vce.ac.in, [4]roopa509@gmail.com, [5]penubakabalaji.cse@gmail.com, [6]ravi.4u.kumar@gmail.com

Correspondence should be addressed to Santosh Kumar Ravva : santosh@staff.vce.ac.in

Article Info
Journal of Machine and Computing (https://anapub.co.ke/journals/jmc/jmc.html)
Doi : https://doi.org/10.53759/7669/jmc202505007
Received 23 May 2024; Revised from 16 August 2024; Accepted 04 October 2024.
Available online 05 January 2025.
©2025 The Authors. Published by AnaPub Publications.
This is an open access article under the CC BY-NC-ND license. (http://creativecommons.org/licenses/by-nc-nd/4.0/)

Abstract – As cybercriminals become their tactics, phishing URLs are increasingly operated to exploit unsuspecting users. This leads to notable financial loss and erodes user trust in online systems, influencing businesses and individuals. Though effective in specific scenarios, traditional signature-based and heuristic methods often require help keeping pace with the dynamic of phishing schemes. In this study, we introduce a hybrid model combining Gated Recurrent Unit (GRU) and Convolutional Neural Networks (CNN) to enhance phishing URL detection. Our primary purpose was to utilize both temporal feature extraction through GRU and spatial feature extraction using CNN, building a robust model capable of effectively identifying phishing attempts. We evaluated three models, GRU, CNN, and the proposed GRU+CNN hybrid, employing a Kaggle dataset containing over 2.5 million URL samples labeled as phishing. The GRU model reached 97.8% accuracy, while the CNN model performed slightly better, with 98% accuracy. However, the hybrid GRU+CNN model outperformed, achieving an accuracy of 99.0%, showing its superiority in addressing the complexities of phishing detection. Future work will optimize the hybrid model for real-time detection and investigate its adaptability to other cybersecurity domains, such as malware and social engineering threats.

Keywords – Phishing URL Detection, Cybersecurity, Hybrid Model, Cyber Threat Prevention, Deep Learning, Phishing Attacks.

I. INTRODUCTION

Since the late 20th century, our society has experienced a significant transformation by technological advancement. Face-to-Face communication has progressively transitioned to digital interactions, with the World Wide Web (WWW) becoming a blessing in our daily lives [1]. Considerable purposes are fulfilled, including instant messaging, online banking, and social media connectivity. Unfortunately, there are vulnerabilities that unscrupulous actors can exploit due to this heavy reliance on the internet [2]. Cybercriminals often use weak security protocols to breach newly implemented systems. Their attacks can potentially result in private data theft, elevating identity theft above and beyond simple financial losses as a severe crime. Attackers can gain unauthorized access to confidential data by attracting users with deceptive URLs (Universal Resource Locators) [3]. Therefore, safeguarding access to URLs has appeared as a vital priority in the fight against cyber threats. As the landscape of cybercrime develops, it is imperative to create robust detection methods to keep pace with cyber adversaries' changing tactics. Our daily activities now rely on the Internet, including communication, business, marketing, education, travel, and shopping. Due to the rise in Internet use and the tendency to share personal information online, sensitive data is more vulnerable to cybercrime. The Internet has many benefits, but criminals have also used it to

commit different crimes, most notably phishing [4]. In these attacks, phoney messages that appear to be from reliable sources are sent via text messages, chat apps, or emails [5].

The term "phishing" is increasingly used and discussed in academic literature, social media, and traditional media. Cybercriminals trick consumers into revealing personal information through various mediums, including phone calls, text messages, chat apps, forums, emails, and URLs [6]. Phishing is a type of cyber-attack that uses social engineering methods [7]. The main objective is to steal confidential information, including passwords, login credentials, and credit card information. Generally, when an attack occurs, a fake domain resembles a legitimate firm's website. After designing fraudulent sites, they sent mass emails to suspects asking them to click on malicious links. These emails often hold threatening or alarming messages prepared to prompt action, eventually leading recipients to a fake login page similar to a trusted site. One key organization monitoring and reporting phishing activities globally is the Anti-Phishing Study Group, which keeps the public informed. The increasing number of phishing attempts, which exceeded 300,000 in July 2023 alone, highlights the urgency for robust cybersecurity measures [8]. In particular, Phishing incidences have expanded from 600 to 1,100 per month across big organizations, the continuing threat with webmail services becoming prime targets. Reports on phishing and fraud indicate a 220% growth in these attacks during the COVID-19 pandemic, demonstrating a marked surge in malicious activity [9]. In a common phishing situation, criminals try to obtain personal information such as login details and financial information for fraud.

Cybercriminals frequently create counterfeit versions of legitimate websites, particularly targeting financial institutions [10]. This tactic attracts victims to fake websites, increasing the risk of data misuse. Many organizations still require sophisticated solutions to identify fake URLs that leave consumers vulnerable, even in the face of the growing requirement for effective phishing prevention [11]. Phishing attacks have increased, especially after the COVID-19 outbreak. As a result, many anti-phishing programs have been developed, also many rely on blocklist databases. Blocklists are a widespread tool used by lookup-based security toolbars in web browsers. These databases are slow to update, leaving users vulnerable to further phishing URLs. Although blocklists seek low false positives, they struggle to keep up with the ongoing creation of new URLs, which limits their effectiveness. Blocklisting is still the most popular way to identify phishing sites in contemporary browsers. However, it is not always effective in identifying newly developed phishing websites not yet included in any blocklist [12].

The shortcomings of conventional techniques have directed researchers to focus on Machine-Learning (ML) algorithms to identify malicious URLs. Modern detection systems have to advance beyond traditional ways as phishing attacks evolve. More dynamic capabilities are available with ML and deep learning (DL) models like Graph Convolutional Networks (GCN), Artificial Neural Networks (ANN), and Recurrent Neural Networks (RNN). These techniques extract features from URLs, create training sets, and apply supervised learning to classify URLs, with state-of-the-art models like XGBoost efficiently handling large datasets for phishing detection. Although these models have limits, they have proved indispensable in detecting and stopping phishing attempts, mainly in digital forensics. One significant drawback is that standalone models are less flexible to threats that appear in real time since they may require assistance with recently created or heavily disguised URLs.

The following are the primary contributions of this paper:

1. We propose a unique combination of GRU and CNN architectures, using GRU's temporal processing strengths with CNN's spatial feature extraction capabilities, presenting a robust and efficient phishing URL detection framework.
2. Our proposed hybrid model outperforms GRU and CNN models, performing an accuracy of 99.0%, incomparable to GRU's 97.8% and CNN's 98%. This demonstrates enhanced capability in phishing URL classification, and the model mitigates overfitting.
3. We evaluate the model on a large Kaggle dataset with over 2.5 million labeled URLs, and the proposed model demonstrates its ability to scale effectively while maintaining high performance and reliability.

This study is divided into multiple sections. Section II will analyze the existing methods for identifying phishing URLs. Section III will describe our proposed method and briefly discuss the configuration for implementing our model. Section IV will provide an overview of the outcomes of the experiments conducted using the models we have proposed. Finally, we will present the findings of our research.

II. LITERATURE REVIEW

Phishing attacks, a severe cybercrime involving email deception and fraudulent websites, threaten internet security. Many research studies have investigated ML methods to identify and prevent phishing attacks, utilizing datasets containing attributes of both phishing and legitimate URLs and commonly utilized ML and ensemble models. The hybrid LSD model [13], which combines LR, SVM, and DT using voting techniques, has shown impressive accuracy. Techniques such as canopy feature selection, cross-validation, and hyperparameter optimization further improve performance, achieving an accuracy of 98.12%. Advancements in phishing URL detection have been made with hybrid deep learning models that integrate NLP features and character embedding-based features. Another study [14] introduced two models, DNN–LSTM and DNN–BiLSTM, which combine high-level NLP features with in-depth character-level analysis. The DNN–BiLSTM model, leveraging the bidirectional LSTM's ability to capture both forward and backward connections, achieved a remarkable 99.21% accuracy on a newly developed dataset, outperforming other models. These models outshine traditional

machine learning and CNN models by utilizing multiple feature sets simultaneously. Future enhancements could further boost the models' performance by incorporating word embeddings and self-attention mechanisms.

Conventional methods for identifying phishing attacks struggle to detect zero-hour attacks and are often unsuitable for real-time environments. To tackle these challenges, a study has suggested a client-side hybrid anti-phishing solution that uses URL and hyperlink features without relying on third-party sources. The study [15] created a new dataset comprising 6000 websites and utilized machine learning models, with the XGBoost classifier achieving an accuracy of 99.17%. The hybrid approach significantly enhances phishing detection, although future endeavours could focus on mobile phishing, an emerging threat. The latest developments in phishing detection involve using advanced deep learning models such as ResNeXt combined with GRU to identify real-time attacks swiftly. The author [16] has incorporated SMOTE to tackle data imbalance and has employed autoencoders with ResNet for more effective feature extraction. The innovative RNT model, further fine-tuned using the Jaya optimization method, has demonstrated exceptional performance on phishing datasets, achieving an impressive 98% accuracy and surpassing state-of-the-art algorithms by 11% to 19%. Its low false positive rates and rapid execution times highlight its efficiency. This approach significantly enhances digital forensics and cybersecurity, particularly in safeguarding wireless communications and administrative data.

Identifying malicious URLs often imitates legitimate ones and presents a significant cybersecurity challenge. Machine learning models have been created to categorize URLs as safe or harmful, safeguarding against phishing attacks and malware. Current research focuses on integrating these models into web browser extensions to alert users promptly. A recent study [17] achieved an accuracy rate of 85.37%, with precision and recall scores of 86.65% and 83.95%, respectively. Although developing a hybrid model presented challenges, potential enhancements, such as incorporating deep learning techniques, may improve detection accuracy and offer more robust real-time protection. The detection of phishing URLs has been significantly improved by integrating Variational Autoencoders (VAE) and DNN in a groundbreaking approach that enhances extracting features from raw URLs. The VAE component plays a crucial role in reconstructing input URLs, enhancing detection accuracy and making the model more effective in identifying malicious URLs [18]. By utilizing datasets such as ISCX-URL-2016 and Kaggle, the proposed model has achieved an impressive accuracy of 97.85% with a rapid response time of 1.9 seconds. These results surpass those of previous models, demonstrating the efficiency and precision of combining VAE and DNN for phishing detection and highlighting its potential for real-time cybersecurity applications.

Phishing detection has made significant strides by developing a layered classification model incorporating URL structure, text, and image features. Unlike previous research, the study [19] leverages a substantial dataset of 20,000 websites, extracting 22 crucial URL features and applying NLP for text analysis. Additionally, text embedded within images, a common characteristic of phishing websites, was processed for classification. The experimental findings revealed that XGBoost outperformed other models, achieving 94% accuracy in training and 91% in testing. MLP and RF also demonstrated high accuracy. This approach notably enhances early phishing detection, bolstering internet security for users. A recent study delves into phishing detection systems that assess request features to pinpoint malicious activities. As the number of features expands, selecting the most relevant ones becomes crucial for enhancing model performance, curbing computational costs, and preventing overfitting. The research [20] strengthens the efficiency of a URL-based phishing detection system by integrating a feature selection method based on GA. Genetic algorithms imitating natural selection were fine-tuned to pinpoint the most pertinent features. The experimental findings reveal that the proposed model attained a 93% accuracy rate, narrowly trailing XGBoost by 1%. This approach presents an encouraging avenue for fine-tuning machine learning models in phishing detection.

Another research [21] presented a hybrid classification model that combines CNN and LSTM algorithms to improve phishing detection, using a substantial dataset of one million URLs and over 10,000 images, the model's sensitivity was evaluated through various factors, including feature types and misclassification rates. A comprehensive experimental analysis demonstrated that IPDS gained an accuracy rate of 93.28% with an average detection time of 25 seconds. This study highlights the effectiveness of deep learning techniques in identifying phishing web pages and attacks in large datasets, significantly contributing to cybersecurity efforts.

III. METHODOLOGY

This study presents a novel Hybrid DL framework, NeuralUnit designed to detect phishing URLs. The growing complexity of phishing attacks, primarily through malicious URLs, necessitates advanced detection mechanisms. To address this, we employ a CNN combined with a GRU to enable reliable feature extraction from the linguistic and sequential patterns in phishing URLs. **Fig 1** graphically represents the overall methodology of our research work.

Dataset Description and Preprocessing

We employed an extensive phishing URL detection Kaggle dataset with over 2.5 million URL samples to evaluate the proposed phishing detection models. This dataset has 18 attributes pertinent to phishing detection tasks and is labelled authentic or phishing. It was assembled using reliable sources like OpenPhish-Community, PhishTank, and Phishing: database and URL rating websites like The Majestic Million and Cisco Umbrella Popularity List. This dataset addresses the lack of sandboxing technologies and big-tagged datasets required for reliable phishing detection. It provides rich context by featuring URLs' linguistic and behavioural characteristics and allowing more accurate detection models. In particular,

the dataset captures crucial features such as URL length, entropy, digit-letter ratios, and domain-related properties that are instrumental for classification algorithms. **Fig 2** displays the dataset sample, and **Table 1** presents the dataset's features.

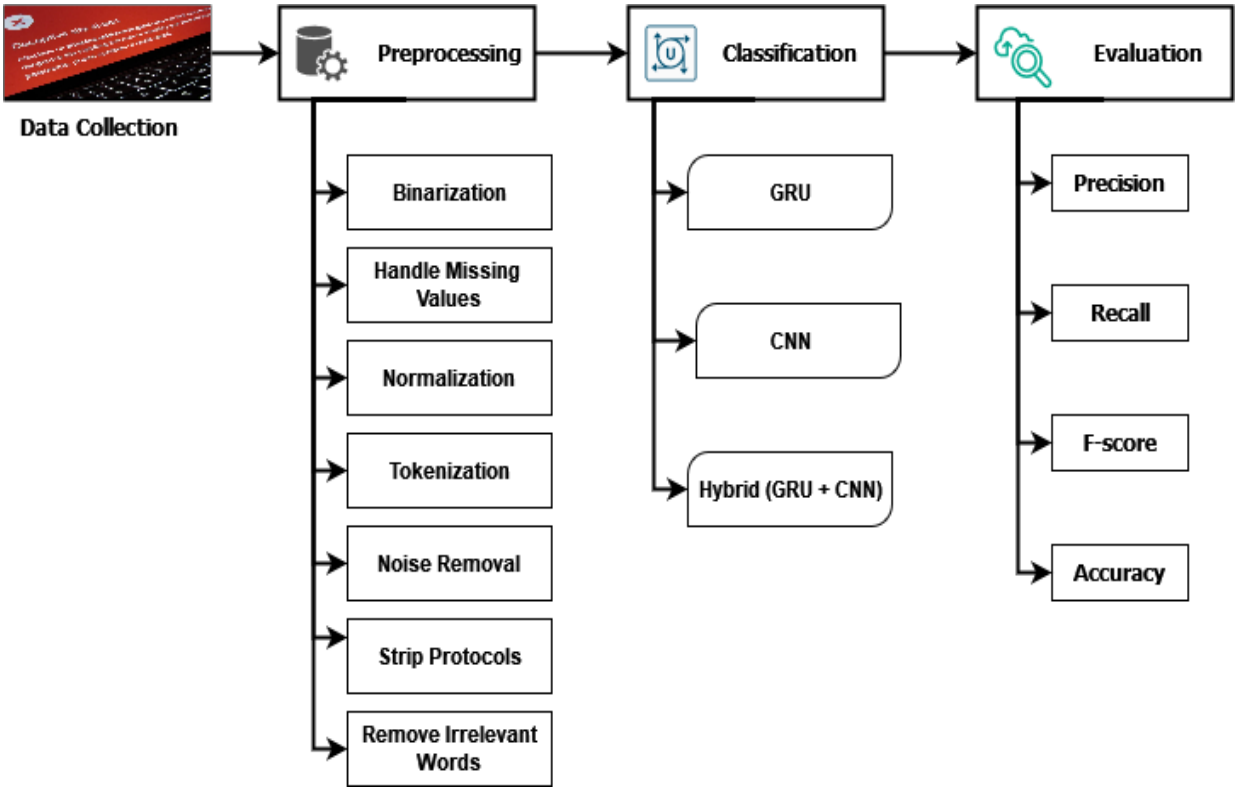

Fig 1. Overview of The Research Methodology Framework.

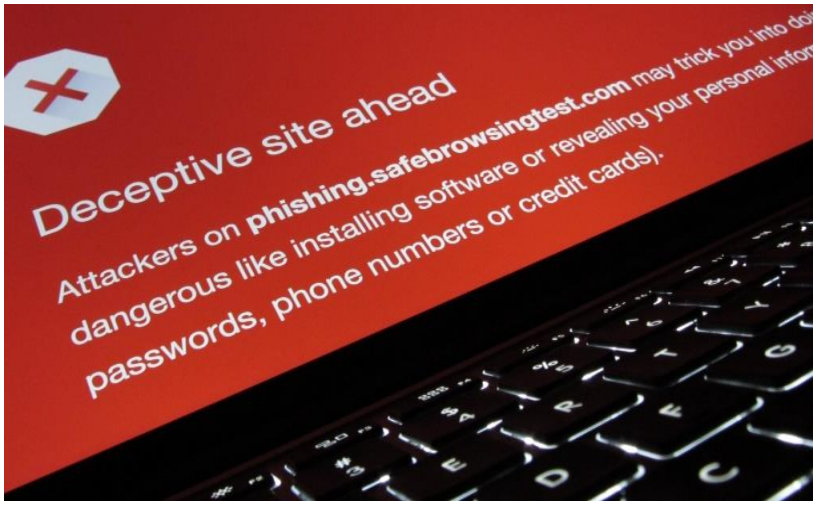

Fig 2. Sample of the Dataset.

Then, some data preprocessing steps were applied. The dataset preprocessing involved several essential steps. Target labels for phishing and legitimate URLs were binarized (1 for phishing, 0 for legitimate), while missing values were appropriately addressed. Continuous features experienced normalization through min-max scaling, and categorical attributes were transformed employing one-hot encoding. The `URL` strings were tokenized into domain and subdomain components to facilitate effective processing within the hybrid CNN-GRU model. Special characters such as "#", "%", and "!" which have no significance for classification, must be removed through data cleaning and noise reduction. Additionally, protocols like "http://", "https://", and "www." are removed, and URLs are changed to lowercase for uniformity. After that, phrases or words that are frequently present in URLs like "home," "index," or "php" are eliminated to concentrate on more pertinent signs of phishing. The dataset has an identical class-wise distribution, with 1,250,000 samples for the legitimate class and 1,250,000 samples for the phishing class. Equitable representation for both classes is guaranteed by this balanced dataset, which is essential for building objective models.

Table 1. Features of the Dataset

#	Column Name	Feature Description	Data Type
1	url	URL string	Str
2	Source	Source of the URL	Str
3	Label	Category of URL, either phishing or legitimate	str
4	url_length	URL length in characters	int
5	starts_with_ip	Is the base URL an IP address?	bool
6	url_entropy	URL/hostname entropy	float
7	has_punycode	Does the URL contain at least one punycode character?	bool
8	digit_letter_ratio	Digit-letter character ratio in URL	float
9	dot_count	Count of occurrences of (.) inside URL	Int
10	at_count	Count of occurrences of the "@" symbol in the URL	Int
11	dash_count	Count of occurrences of "-" symbol in URL	Int
12	tld_count	Does the subdirectory in the URL contain TLDs?	int
13	domain_has_digits	Does the domain (base URL) contain digits?	bool
14	subdomain_count	Count of subdomains in the base URL	Int
15	nan_char_entropy	Character entropy of non-alphanumeric characters	float
16	has_internal_links	Does the URL subdirectory contain internal links?	bool
17	whois_data	Availability of domain WHOIS record	bool
18	domain_age_days	Domain age in days (based on WHOIS data)	float

Dataset Statistical Information

A histogram plot showing the distribution of the url_entropy feature among the two classes (phishing and legitimate) is shown in **Fig 3**. Hue='label' is used in the graphic to differentiate the two classes so that the URL entropy distributions of each may be compared directly. The underlying probability distribution of the data is also represented more smoothly by the overlay of the Kernel Density Estimate (KDE) curve. This graphic illustrates the differences in URL entropy between phishing and authentic URLs, maybe pointing forth trends that could be helpful for categorization.

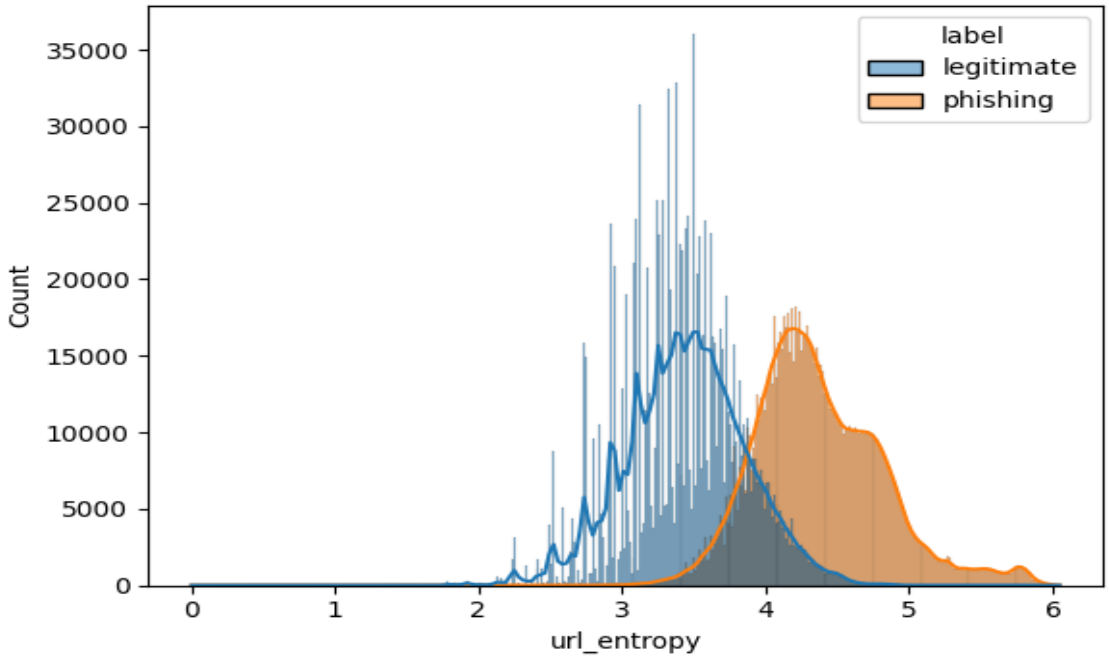

Fig 3. Distribution of url_entropy for Legitimate and Phishing URLs, with KDE Curves Providing a Smooth Data Representation Across Both Classes.

As seen in **Fig 4**, the plot displays the distribution of the url_length characteristic for both phishing and authentic URLs. The graphic distinguishes between the two classes using hue='label,' making it possible to compare URL lengths between phishing and genuine websites. Including the Kernel Density Estimate (KDE) curve offers a smooth depiction of the distribution. Based on this characteristic, this visualization aids in highlighting any variations or patterns in URL length between the two groups, which may help identify phishing URLs.

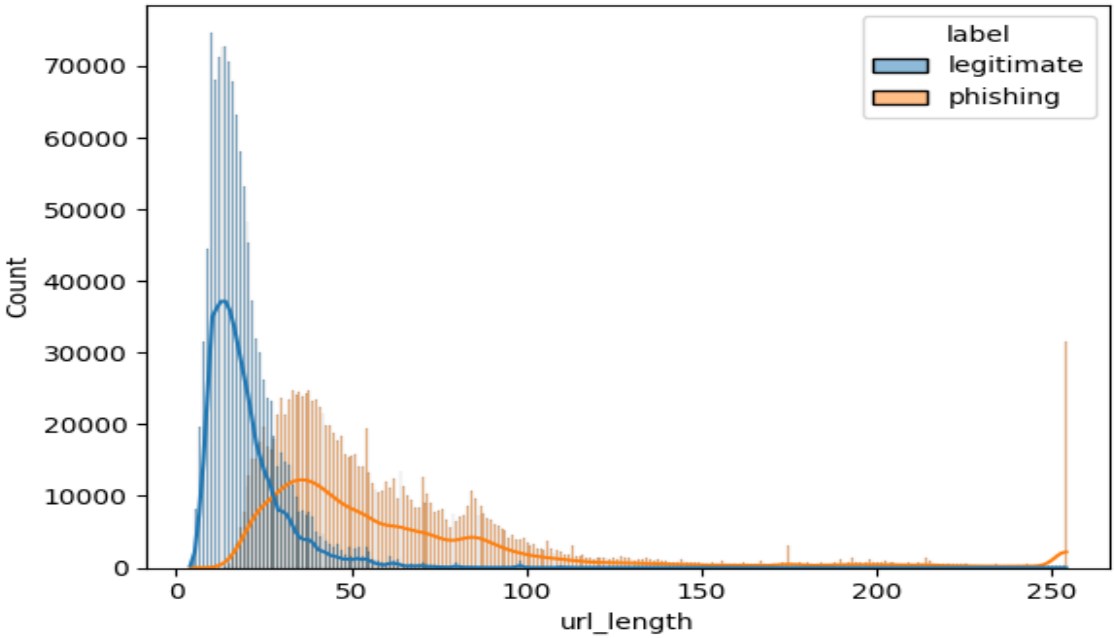

Fig 4. The Distribution of url_length for Phishing and Valid URLs; the KDE Curves Indicate a Smooth Representation of the Data for Both Classes.

The distribution of the subdomain_count feature for phishing and authentic URLs is shown in **Fig 5**. To differentiate between the two groups, the hue='label' parameter is employed, and the KDE curve is incorporated to offer a smooth approximation of the underlying probability distribution. The number of subdomains found in authentic versus phishing URLs can be compared using this graphic. The plot assists in identifying possible variations in URL structures by examining the distribution of subdomain counts. Phishing URLs may have more or fewer subdomains than authentic ones, which can be a significant indicator for classification models.

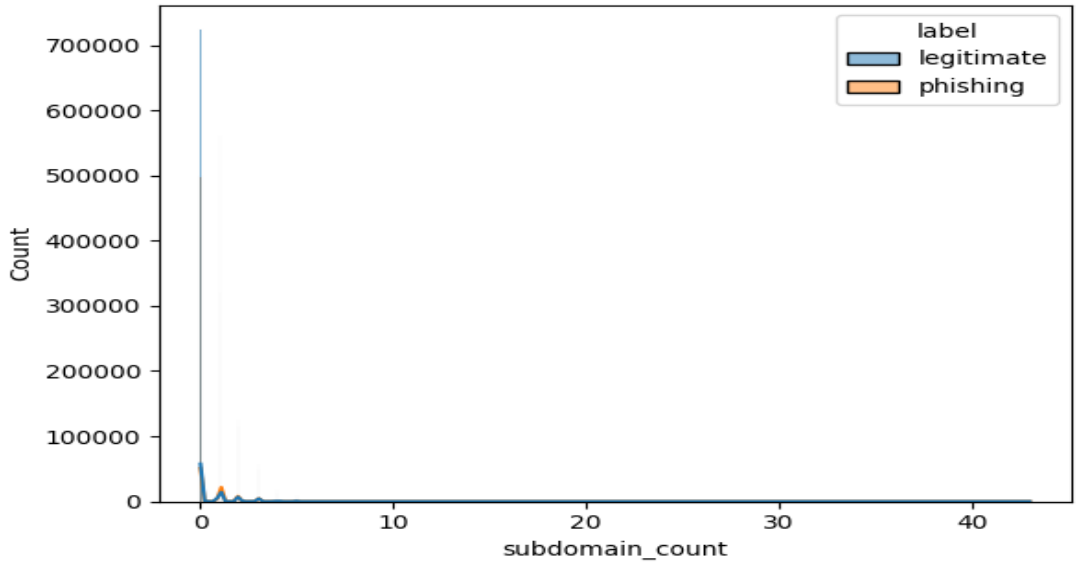

Fig 5. The Distribution of subdomain_count for Phishing and Authentic URLs. KDE Curves Show How the Data is Represented Smoothly in Both Groups.

Gated Recurrent Unit (GRU) Model

The GRU is an efficient variant of the Long Short-Term Memory (LSTM) network, created to sweeten computational performance by simplifying its architecture. In contrast to the three-gate structure of the LSTM, the GRU creates a two-gate system with a reset gate and combines the forget and input gates into a single update gate [22]. The update gate and reset gate are the two main computing components of a GRU. Initially, the update gate u_t determines the amount of the previous concealed state to be kept for the current time step, which is computed as follows:

$$u_t = \sigma\,(W_u.x_t + U_u.h_{t-1} + b_u) \tag{1}$$

where W_u and U_u are the weight matrices for the input x_t and the prior hidden state h_{t-1}, respectively, and b_u is the bias term. The function σ defines the sigmoid activation function:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2}$$

Moving to the reset gate r_t, which prescribes how much of the prior hidden state should be ignored, is calculated as:

$$r_t = \sigma\,(W_r.x_t + U_r.h_{t-1} + b_r) \tag{3}$$

where W_r, U_r, and b_r are the weight matrices and bias for the reset gate.

Once the reset gate is used, the candidate hidden state $\hat{h}_t$ is produced by considering both the current input and the adjusted hidden state from the previous time step:

$$\hat{h}_t = tanh\,(W_h.x_t + r_t * (U_h.h_{t-1})) \tag{4}$$

where W_h and U_h are weight matrices, and (*) represents element-wise multiplication. The function $tanh$ expresses the hyperbolic tangent function that introduces non-linearity.

Finally, the hidden state h_t at the current time step is updated between the candidate hidden state $\hat{h}_t$ and the prior hidden state h_{t-1}, with the update gate u_t :

$$h_t = (1 - u_t) * h_{t-1} + u_t * \hat{h}_t) \tag{5}$$

The output y_t for the current time step is estimated based on the hidden state:

$$y_t = \sigma\,(W_y.h_t + b_y) \tag{6}$$

where W_y is the output weight matrix, and b_y is the output bias. **Fig 6** illustrates the GRU structure.

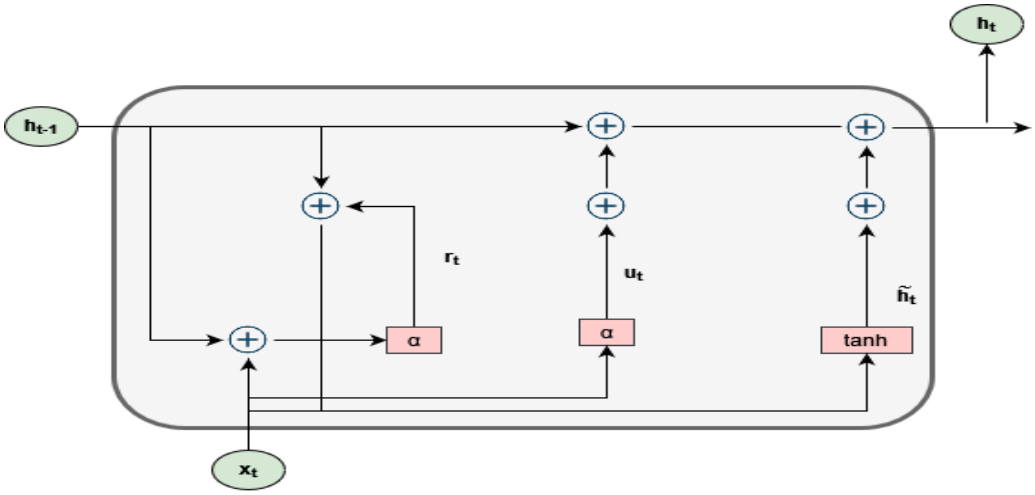

Fig 6. GRU Model Architecture.

Convolutional Neural Network (CNN) Model

CNNs represent a powerful class of deep neural networks, predominantly employed in tasks such as image recognition, video analysis, and time series classification. Their strength lies in their capability for effective feature extraction and spatial hierarchy learning, allowing them to tackle complex problems remarkably. A typical CNN design consists of numerous convolutional layers, pooling layers, fully connected layers, an output layer, and an input layer. The input layer receives and preprocesses data before it is sent over the network by associating every neuron with a feature in the incoming data; this layer ensures that the dimensions are matched accurately. Using several kernels to create new feature maps and appending a bias term after each convolution, the convolutional layers perform convolution operations on the input data. This can lead to high-dimensional feature maps, requiring a decrease in dimensionality to mitigate overfitting and improve learning efficiency. Pooling techniques, such as max pooling and average pooling, are utilized for this purpose [23].

The fully linked layers are critical in combining local features into global representations for high-level inference. Mathematically, the output of the fully connected layer for the j[th] neuron is stated as follows:

$$o_j = \sum_{k=1}^{m} W_{jk} a_k + b_o \tag{7}$$

where o_j represents the output of the jth neuron, m is the number of input features a_k, W_{jk} defines the weight connecting the k[th] input to the j[th] neuron, and b_o is the connected bias term.

For activation, this model employs the Rectified Linear Unit (ReLU) function, described as:

$$a_j = g(o_j) = max(0, o_j) \tag{8}$$

In this context, a_j denotes the activated output for the j[th] neuron, effectively presenting non-linearity into the network and enabling it to learn complex patterns. **Fig 7** represents the CNN model architecture.

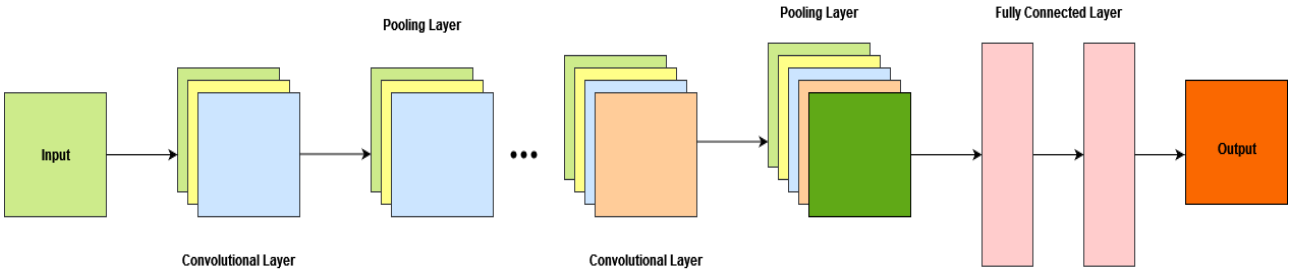

Fig 7. CNN Model Architecture.

Proposed Model

We introduce a novel hybrid method that Combines the strengths of CNN and GRU to improve phishing URL detection. This architecture is developed to extract spatial features from URLs while capturing sequential dependencies, improving detection accuracy. The model initiates with a CNN that processes the input URLs. The convolutional layers apply filters to extract meaningful features, and after being flattened, the output of the final convolutional layer is transmitted to the fully connected layer. The following represents the output of this layer:

$$y_j = i = 1nW_{ij}x_i + b1 \tag{9}$$

where y_j defines the output of the j[th] neuron, W_{ij} denotes the weights connecting the i[th] input to the j[th] neuron, x_i are the flattened features, and $b1$ is the bias term.

Following feature extraction, the GRU component processes the sequential output from the CNN. The GRU utilizes an update gate z_t and a reset gate r_t to manage information flow. These gates are calculated as follows:

$$z_t = \sigma\, W_z x_t + U_z h_{t-1} + b_z \tag{10}$$

$$r_t = \sigma\, W_r x_t + U_r h_{t-1} + b_r \tag{11}$$

where the prior hidden state is denoted by h_{t-1}, the current input is indicated by x_t, the update and reset gate outputs are represented by z_t and r_t, respectively, and the weight matrices for the gates are defined by W and U.

The hidden state h_t of the GRU is updated using:

$$h_t = (1 - z_t) * h_{t-1} + z_t * tanh\,(W_h x_t + U_h\,(r_t * h_{t-1}) + b_h) \tag{12}$$

This mechanism allows the model to retain relevant information from previous time steps while learning to forget unnecessary details.

Finally, the output of the GRU is fed into a classification layer to predict whether a URL is phishing or legitimate. The prediction can be depicted as follows:

$$\hat{y} = \sigma\,(W_{out} h_T + b_{out}) \tag{13}$$

where $\hat{y}$ is the predicted probability, h_T is the hidden state at the final time step, W_{out} is the weight matrix for the output layer, and b_{out} is the output bias.

Our proposed hybrid model, which combines CNN and GRU, improves feature extraction and detects sequential patterns in URL data, enhancing phishing detection performance. This method addresses the challenges of recognizing malicious URLs in a complex threat landscape. **Fig 8** depicts the overall architecture of our proposed Hybrid model.

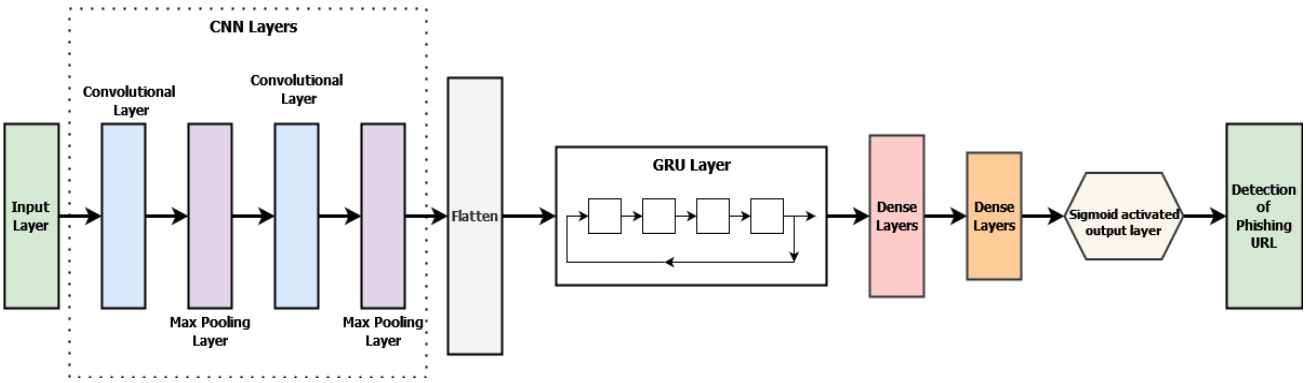

Fig 8. Graphical Representation of Our Proposed Hybrid Model.

IV. RESULT AND DISCUSSION

This section evaluates the proposed hybrid framework's performance compared to DL models and widely used ML algorithms. The evaluation was conducted through experiments on various well-known SMS phishing datasets.

Experimental Setup

After collecting the datasets, we developed the models using the algorithms discussed in the previous section. We utilized Google Colab Pro, a subscription-based platform, to access the necessary computational resources for training. Since the selected models and datasets required GPU support, we opted for V100 GPUs for our experiment. The dataset, consisting of 1,250,000 samples for legitimate and phishing classes, was divided utilizing the train_test_split method. Around 70% of the data was allocated for training, while the remaining 30% was for testing, ensuring. The following sections will present different models' evaluation parameters and performance results.

Evaluation Metrics

Evaluating the ML model's performance with clear and robust metrics is crucial for validating its effectiveness. To assess the success of our framework, we employed the following performance indicators.

Accuracy

Accuracy is calculated as the sum of true positives (TP) and true negatives (TN) divided by the total number of predictions made (true positives, true negatives, false positives, and false negatives). The formula represents this:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{14}$$

Recall

Recall refers to the proportion of actual positive instances that are correctly identified. It measures the model's ability to capture legitimate SMS and is defined as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{15}$$

Precision

Precision indicates the fraction of correctly predicted positive samples out of all instances classified as positive by the model. It can be represented as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{16}$$

F-score

The f-score balances the trade-off between precision and recall, defined as the harmonic mean of these two metrics:

$$\text{F1- score} = 2x \frac{Recall\ x\ Precision}{Recall + Precision} \tag{17}$$

Performance of models

Table 2. Training, Testing, and Validation Accuracy of Different Models

Models	Training Accuracy	Testing Accuracy	Validation Accuracy
GRU	98.2%	97.8%	97.3%
CNN	98.5%	98%	97.8%
Proposed Hybrid (GRU+CNN) Model	99.2%	99%	98.9%

Table 2 comprehensively compares all evaluated models, focusing on their training, testing, and validation accuracy, which depicts the performance of each model, GRU, CNN and proposed Hybrid (GRU+CNN) throughout the training, testing (indicating their ability to generalize to unseen data), and validation phases (measuring their consistency). Notably, the hybrid model (GRU+CNN) demonstrated exceptional accuracy across all stages, yielding near-perfect results due to its superior fault detection capabilities. We compared GRU, CNN, and a hybrid model (GRU+CNN) performance for detecting phishing attacks. The GRU model achieved a training accuracy of 98.2%, a testing accuracy of 97.8%, and a validation accuracy of 97.3%. Although the GRU model demonstrated strong learning capability, the marginal decline in testing and validation accuracy indicates some limitations in fully generalizing the patterns in phishing data.

On the other hand, the CNN model exhibited slightly superior performance, with a training accuracy of 98.5%, testing accuracy of 98.0%, and validation accuracy of 97.8%. This underscores CNN's effectiveness in capturing spatial features, leading to enhanced generalization. However, as evidenced by the proximity of testing and validation scores to those of GRU, it still falls short of being optimal for phishing detection.

Conversely, the proposed hybrid model that integrates GRU and CNN surpassed the individual models, achieving a training accuracy of 99.2%, testing accuracy of 99.0%, and validation accuracy of 98.9%. This substantial enhancement underscores how leveraging the combined strengths of GRU's sequential learning and CNN's feature extraction capabilities enabled the model to comprehend better and generalize complex phishing data. This hybrid approach yielded the most favourable results, emphasizing its suitability for robust phishing detection tasks, offering superior accuracy and generalization capabilities in testing and validation.

Table 3. Precision, Recall, and F1-Score for Each Model

Models	Precision	Recall	F-score
GRU	95.6%	90.2%	92.5%
CNN	98.5%	84.8%	90.4%
Proposed Hybrid (CNN+GRU)	96.5%	93.0%	95.2%

Table 3 displays a comparison of the performance of the GRU, CNN, and the combined CNN+GRU models' performance. The comparison is based on precision, recall, F1-score, and accuracy. The GRU model demonstrates a high precision of 95.6% and a recall of 90.2%, resulting in an F1-score of 92.5%, indicating its proficiency in identifying relevant patterns. However, the relatively lower recall suggests that it still misses some potential positive cases, potentially leading to false negatives. On the other hand, the CNN model boasts an impressive precision of 98.5% but has a lower recall of 84.8%, indicating its struggle in identifying all true positives.

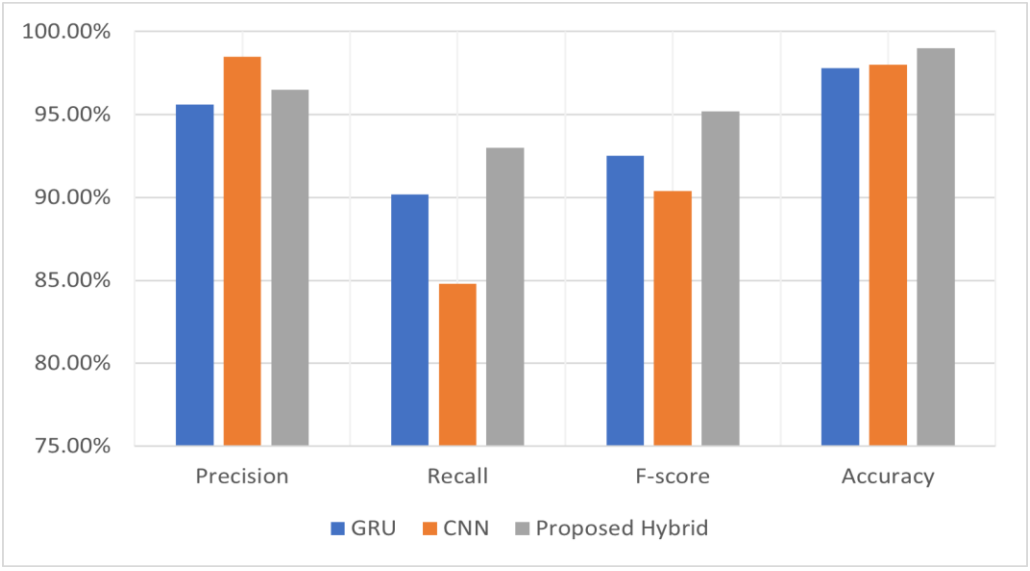

Fig 9. Comparative Performance of GRU, CNN, and Proposed Hybrid Models.

The hybrid CNN+GRU model significantly enhances performance across all metrics, achieving 96.5% precision, 93.0% recall, and a superior F1 score of 95.2%. The balanced and higher scores of the combined model suggest that integrating both CNN and GRU compensates for the individual weaknesses of each model, leading to a more reliable and generalizable solution for fault detection tasks. This comprehensive improvement in all metrics demonstrates that the hybrid model effectively addresses the complexities inherent in phishing URL detection, similar to how combining various models can overcome individual limitations.

Fig 9 displays that while GRU and CNN models perform well, they have individual limitations in balancing precision and recall. Compared to CNN, the GRU model exhibits some shortcomings in precision, but it performs exceptionally well when processing sequential data. On the other hand, the CNN model provides excellent precision but needs help with recall, implying challenges in capturing complex relationships. Our proposed hybrid model, which incorporates the strengths of GRU and CNN, addresses these shortcomings by using the sequential learning capabilities of GRU and the powerful feature extraction of CNN. This synergy results from a more evenly distributed performance across all indicators. The benefit of this hybrid technique is that it can increase overall accuracy and dependability in identifying intricate patterns, making it a more reliable solution.

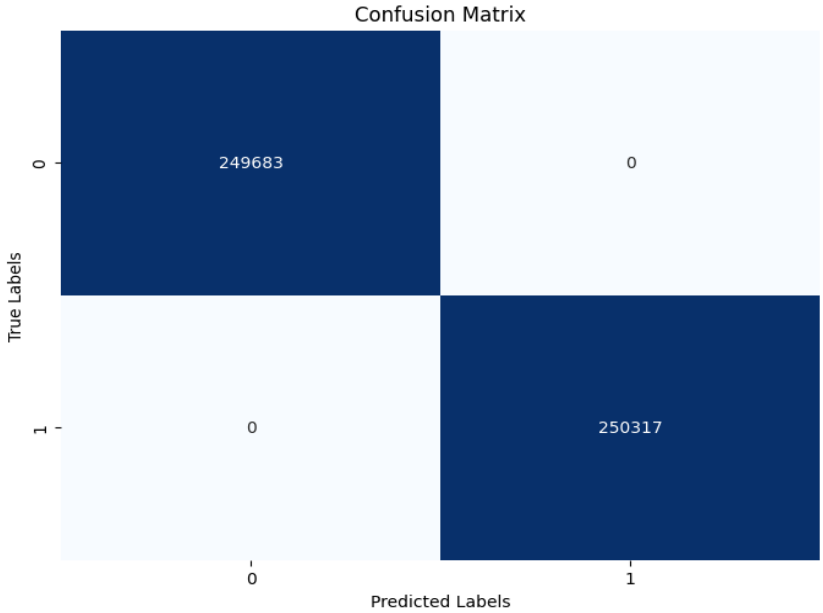

Fig 10. Confusion Matrix of The Proposed Hybrid Model.

Fig 10 shows the confusion matrix is used to examine the faults in the suggested model. By displaying the true positive, true negative, false positive, and false negative counts for each class, it offers a thorough analysis of the model's performance. This matrix makes it possible to assess the model's performance in differentiating between phishing and authentic URLs and shows places where misclassifications happen. We can pinpoint particular shortcomings in the model's predictions and direct future enhancements to its functionality by looking at the confusion matrix.

Discussion

Table 4 illustrates the comparative analysis, which shows significant enhancements achieved by our method, especially in attaining greater accuracy and a more balanced performance evaluation. This comparison provides strong support for the progress of our research, highlighting the effectiveness and originality of our approach compared to previous studies. Compared to existing models in the literature, the comparative analysis of the proposed GRU+CNN model underscores its superior performance in detecting phishing URLs.

Table 4. Comparative Performance Analysis with Existing Work

Reference	**Model**	**Accuracy**
Ariyadasa et. al [24]	LSTM+CNN	98.34%
Shaukat et. al [25]	Linear SVC	98.9%
Choudhury et. al [26]	XGBoost	96.9%
Adebowale et. al [27]	CNN	92.5%
Proposed Model	GRU+CNN	99.0%

The table illustrates that the GRU+CNN model achieves an accuracy of 99.0%, surpassing the accuracy rates of several notable models. For example, Shaukat et al. [25] reported an accuracy of 98.9% with a Linear SVC model, while Ariyadasa et al. [24] achieved 98.34% with an LSTM+CNN architecture. Although these models demonstrate commendable performance, they do not match the accuracy attained by the proposed model.

Furthermore, Choudhury et al. [26] found that their XGBoost model recorded an accuracy of 96.9%, and Adebowale et al. [27] reported 92.5% for their CNN approach. These figures indicate that while various advanced models have been explored in fault detection, none match the efficacy of the GRU+CNN hybrid framework. The improved accuracy of the proposed model suggests a significant advancement in the ability to reliably detect machinery faults, making it a valuable contribution to predictive maintenance. This enhanced performance is likely attributable to the synergistic integration of GRU and CNN, which effectively captures temporal features.

V. CONCLUSION

The majority of cyberattacks in cyberspace today use phishing messages, which are becoming more prevalent. While most researchers provide sophisticated techniques to reduce the frequency of these assaults, more work has to be done. This paper introduces a cutting-edge hybrid model that integrates the strengths of CNN and GRU for phishing URL detection. By combining CNN's ability to extract complicated features from URL structures with GRU's capacity to capture sequential dependencies, our model shows a slight understanding of the patterns that differentiate phishing URLs from legitimate ones. The integration of these two architectures, along with the use of pre-trained embeddings, results in a comprehensive feature extraction process that improves detection accuracy and robustness. The model addresses the limitations of individual approaches and demonstrates high efficiency in handling large-scale URL datasets. The outcomes suggest that this hybrid architecture significantly advances phishing detection methodologies, providing a scalable and reliable solution in the evolving landscape of cybersecurity. Future research could fine-tune the model's components and examine cross-domain adaptability to strengthen its applicability.

CRediT Author Statement

The authors confirm contribution to the paper as follows:
Conceptualization: Sangeetha M, Navaz K, Santosh Kumar Ravva, Roopa R, Penubaka Balaji and Ravi Kumar T; **Methodology:** Sangeetha M, Navaz K and Santosh Kumar Ravva; **Software:** Roopa R, Penubaka Balaji and Ravi Kumar T; **Data Curation:** Sangeetha M, Navaz K and Santosh Kumar Ravva; **Writing- Original Draft Preparation:** Sangeetha M, Navaz K, Santosh Kumar Ravva, Roopa R, Penubaka Balaji and Ravi Kumar T; **Visualization:** Sangeetha M, Navaz K, Santosh Kumar Ravva, Roopa R, Penubaka Balaji and Ravi Kumar T; **Investigation:** Navaz K, Santosh Kumar Ravva, Roopa R and Penubaka Balaji; **Supervision:** Roopa R, Penubaka Balaji and Ravi Kumar T; **Validation:** Sangeetha M, Navaz K, Santosh Kumar Ravva, Roopa R, Penubaka Balaji and Ravi Kumar T; **Writing- Reviewing and Editing:** Sangeetha M, Navaz K, Santosh Kumar Ravva, Roopa R, Penubaka Balaji and Ravi Kumar T; All authors reviewed the results and approved the final version of the manuscript.

Data Availability

No data was used to support this study.

Conflicts of Interests

The author(s) declare(s) that they have no conflicts of interest.

Funding

No funding agency is associated with this research.

Competing Interests

There are no competing interests

References

[1]. C. Lai et al., "URL Phishing Detection by Using Natural Language Processing and Deep Learning Model," New Trends in Intelligent Software Methodologies, Tools and Techniques, Sep. 2024, doi: 10.3233/faia240360.
[2]. M. A. I. Mallick and R. Nath, "Navigating the Cyber security Landscape: A Comprehensive Review of Cyber-Attacks, Emerging Trends, and Recent Developments," World Scientific News, vol. 190, pp. 1--69, 2024.
[3]. S. Roy, "Cyber Deception against Adversarial Reconnaissance in Enterprise Network using Semi-Indistinguishable Honeypot," 2024.
[4]. M. M. Ali and N. F. Mohd Zaharon, "Phishing—A Cyber Fraud: The Types, Implications and Governance," International Journal of Educational Reform, vol. 33, no. 1, pp. 101–121, Mar. 2022, doi: 10.1177/10567879221082966.
[5]. M. Safaei Pour, C. Nader, K. Friday, and E. Bou-Harb, "A Comprehensive Survey of Recent Internet Measurement Techniques for Cyber Security," Computers & Security, vol. 128, p. 103123, May 2023, doi: 10.1016/j.cose.2023.103123.
[6]. M. K. Prabakaran, P. Meenakshi Sundaram, and A. D. Chandrasekar, "An enhanced deep learning-based phishing detection mechanism to effectively identify malicious URLs using variational autoencoders," IET Information Security, vol. 17, no. 3, pp. 423–440, Jan. 2023, doi: 10.1049/ise2.12106.

[7]. G. D. G. Jaime, E. A. de Jesus, and C. M. N. A. Pereira, "EVALUATING PHISHING RISKS AND SOCIAL ENGINEERING IN A NUCLEAR INFRASTRUCTURE: A CYBERSECURITY CASE STUDY," Instituto de Engenharia Nuclear: Progress Report, pp. 52--54, 2024.

[8]. S. Patil and S. Dhage, "A Methodical Overview on Phishing Detection along with an Organized Way to Construct an Anti-Phishing Framework," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), vol. 15, pp. 588–593, Mar. 2019, doi: 10.1109/icaccs.2019.8728356.

[9]. B. C. Ujah-Ogbuagu, O. N. Akande, and E. Ogbuju, "A hybrid deep learning technique for spoofing website URL detection in real-time applications," Journal of Electrical Systems and Information Technology, vol. 11, no. 1, Jan. 2024, doi: 10.1186/s43067-023-00128-8.

[10]. M. Ravi Prasad and N. Thillaiarasu, "Multichannel EfficientNet B7 with attention mechanism using multimodal biometric- based authentication for ATM transaction," Multiagent and Grid Systems, vol. 20, no. 2, pp. 89–108, Aug. 2024, doi: 10.3233/mgs-230118.

[11]. G. Varshney, R. Kumawat, V. Varadharajan, U. Tupakula, and C. Gupta, "Anti-phishing: A comprehensive perspective," Expert Systems with Applications, vol. 238, p. 122199, Mar. 2024, doi: 10.1016/j.eswa.2023.122199.

[12]. F. Carroll, J. A. Adejobi, and R. Montasari, "How Good Are We at Detecting a Phishing Attack? Investigating the Evolving Phishing Attack Email and Why It Continues to Successfully Deceive Society," SN Computer Science, vol. 3, no. 2, Feb. 2022, doi: 10.1007/s42979-022-01069-1.

[13]. A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, "Phishing Detection System Through Hybrid Machine Learning Based on URL," IEEE Access, vol. 11, pp. 36805–36822, 2023, doi: 10.1109/access.2023.3252366.

[14]. A. Ozcan, C. Catal, E. Donmez, and B. Senturk, "A hybrid DNN–LSTM model for detecting phishing URLs," Neural Computing and Applications, vol. 35, no. 7, pp. 4957–4973, Aug. 2021, doi: 10.1007/s00521-021-06401-z.

[15]. S. Das Guptta, K. T. Shahriar, H. Alqahtani, D. Alsalman, and I. H. Sarker, "Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning Techniques," Annals of Data Science, vol. 11, no. 1, pp. 217–242, Mar. 2022, doi: 10.1007/s40745-022-00379-8.

[16]. F. S. Alsubaei, A. A. Almazroi, and N. Ayub, "Enhancing Phishing Detection: A Novel Hybrid Deep Learning Framework for Cybercrime Forensics," IEEE Access, vol. 12, pp. 8373–8389, 2024, doi: 10.1109/access.2024.3351946.

[17]. A. Pandey and J. Chadawar, "Phishing URL detection using hybrid ensemble model," International Journal Of Engineering Research \& Technology (IJERT), vol. 11, 2022.

[18]. Srilatha, Doddi, and N. Thillaiarasu. "Implementation of Intrusion detection and prevention with Deep Learning in Cloud Computing." Journal of Information Technology Management 15.Special Issue (2023): 1-18..

[19]. M. W. Shaukat, R. Amin, M. M. A. Muslam, A. H. Alshehri, and J. Xie, "A Hybrid Approach for Alluring Ads Phishing Attack Detection Using Machine Learning," Sensors, vol. 23, no. 19, p. 8070, Sep. 2023, doi: 10.3390/s23198070.

[20]. E. Kocyigit, M. Korkmaz, O. K. Sahingoz, and B. Diri, "Enhanced Feature Selection Using Genetic Algorithm for Machine-Learning-Based Phishing URL Detection," Applied Sciences, vol. 14, no. 14, p. 6081, Jul. 2024, doi: 10.3390/app14146081.

[21]. M. A. Adebowale, K. T. Lwin, and M. A. Hossain, "Intelligent phishing detection scheme using deep learning algorithms," Journal of Enterprise Information Management, vol. 36, no. 3, pp. 747–766, Jun. 2020, doi: 10.1108/jeim-01-2020-0036.

[22]. X. Gao, X. Li, B. Zhao, W. Ji, X. Jing, and Y. He, "Short-Term Electricity Load Forecasting Model Based on EMD-GRU with Feature Selection," Energies, vol. 12, no. 6, p. 1140, Mar. 2019, doi: 10.3390/en12061140.

[23]. M. Sabri and M. El Hassouni, "A Novel Deep Learning Approach for Short Term Photovoltaic Power Forecasting Based on GRU-CNN Model," E3S Web of Conferences, vol. 336, p. 00064, 2022, doi: 10.1051/e3sconf/202233600064.

[24]. A. et al., "Detecting phishing attacks using a combined model of LSTM and CNN," International Journal of ADVANCED AND APPLIED SCIENCES, vol. 7, no. 7, pp. 56–67, Jul. 2020, doi: 10.21833/ijaas.2020.07.007.

[25]. M. W. Shaukat, R. Amin, M. M. A. Muslam, A. H. Alshehri, and J. Xie, "A Hybrid Approach for Alluring Ads Phishing Attack Detection Using Machine Learning," Sensors, vol. 23, no. 19, p. 8070, Sep. 2023, doi: 10.3390/s23198070.

[26]. T. Choudhary, S. Mhapankar, R. Bhddha, A. Kharuk, and R. Patil, "Machine Learning Approach for Phishing Attack Detection," Journal of Artificial Intelligence and Technology, May 2023, doi: 10.37965/jait.2023.0197.

[27]. M. A. Adebowale, K. T. Lwin, and M. A. Hossain, "Intelligent phishing detection scheme using deep learning algorithms," Journal of Enterprise Information Management, vol. 36, no. 3, pp. 747–766, Jun. 2020, doi: 10.1108/jeim-01-2020-0036.