Journal of Data Analysis and Information Processing, 2025, 13(3), 331-346
https://www.scirp.org/journal/jdaip
ISSN Online: 2327-7203
ISSN Print: 2327-7211

Effectiveness of Deep Learning Algorithms in Phishing Attack Detection for Cybersecurity Frameworks

Mitra Penmetsa[1], Jayakeshav Reddy Bhumireddy[2], Rajiv Chalasani[3], Srikanth Reddy Vangala[4], Ram Mohan Polam[1], Bhavana Kamarthapu[5]

[1]University of Illinois at Springfield, Springfield, IL, USA
[2]University of Houston, Houston, TX, USA
[3]Sacred Heart University, Fairfield, CT, USA
[4]University of Bridgeport, Bridgeport, CT, USA
[5]Fairleigh Dickinson University, Teaneck, NJ, USA
Email: mitravarma.penmetsa@gmail.com, jayakeshav10807@gmail.com, Rajivchalasani555@gmail.com, Srikanthreddy1043@gmail.com, ramreddy.polam@gmail.com, kamarthapubhavana6@gmail.com

How to cite this paper: Penmetsa, M., Bhumireddy, J.R., Chalasani, R., Vangala, S.R., Polam, R.M. and Kamarthapu, B. (2025) Effectiveness of Deep Learning Algorithms in Phishing Attack Detection for Cybersecurity Frameworks. *Journal of Data Analysis and Information Processing*, **13**, 331-346.
https://doi.org/10.4236/jdaip.2025.133021

Received: June 17, 2025
Accepted: August 25, 2025
Published: August 28, 2025

Abstract

The widespread use of internet technologies is limited because people are worried about cybersecurity. With phishing, cyber criminals pose as reputable entities to trick users and access important information. Standard detection approaches are difficult to follow along with the constantly changing strategies of cybercriminals. A new phishing attack detection framework is presented in this research, using the Gated Recurrent Unit (GRU) Artificial Intelligence (AI) model. Labels have been added to the Uniform Resource Locators (URLs) in the PhishTank dataset, so the model learns what is phishing and what is not. A good data preprocessing method involving feature extraction, dealing with missing data, and running outlier detection checks is applied to maintain high data quality. The performance of the GRU model is outstanding, reaching 98.01% accuracy, F1-score of 98.14%, 98.41% recall, as well as 98.67% precision, better than that of classical Machine Learning (ML) methods, including Adaptive Boosting (AdaBoost) and Long Short-Term Memory (LSTM). The proposed approach correctly handles dependencies among elements in a URL, resulting in a strong method for detecting phishing pages. Results from experiments verify the model's potential in accurately identifying phishing attacks, offering significant advancements in cybersecurity defense systems.

Keywords

Cybersecurity, Phishing Attacks, Machine Learning, Deep Learning (DL),

GRU, PhishTank Data, Cyber Attack Defense

1. Introduction

Identity theft, known as "phishing", occurs when criminals pose as reliable businesses to get personal information, including usernames, passwords, credit card numbers, and account details [1]. Typically carried out through emails, Phishing attacks direct people to fake websites that closely mimic authentic ones, deceiving individuals into divulging confidential information. Common targets include banks, online retailers, social media platforms, auction websites, and online payment gateways, including eBay, PayPal, Facebook, YouTube, and Wells Fargo.

Despite the existence of several anti-phishing software solutions and detection techniques, phishers continuously develop new and hybrid strategies to circumvent existing defenses [2]. Traditional rule-based algorithms, while useful, are often insufficient against the evolving complexity and sophistication of phishing attacks. Consequently, researchers have increasingly explored Machine Learning (ML) and Deep Learning (DL) techniques as more robust and adaptive alternatives for phishing detection.

Binary classification is the most often utilized of the three categories of phishing detection techniques that involve ML: multi-class, multi-label, and binary classification [3]. In this case, emails are sorted as spam (phishing) or legitimate (ham). Extracting information from email messages is key to the classification process, and algorithms based on ML provide much better results than text-based methods.

Cybersecurity frameworks are now making more use of ML and DL techniques to enhance what they can detect [4]. Although a fully autonomous definition against cyber-attacks is not possible yet, ML and DL systems in Security Operation Centres and Network Operation Centres have made it much easier to respond to phishing attacks [5]. With this progress in mind, this study studies whether DL algorithms are effective for detecting phishing, supporting the progress of flexible and advanced cybersecurity systems.

1.1. Motivation and Contribution of the Study

Cybercriminals still use phishing as one of their favourite ways to commit fraud and steal key information because they regularly update their methods. Typical detection methods often struggle to recognize smart phishing scams due to their inability to respond to ongoing changes and miss detailed anomalies. Given the growing complexity of phishing techniques, there is an urgent demand for more sophisticated, adaptive, and scalable solutions. DL models show great potential in identifying phishing attempts by recording intricate, temporal dependencies in URL information. This project's goal is to examine how Gated Recurrent Unit (GRU) could improve cybersecurity and phishing detection. The following are this work's

main contributions:

- Utilization of the PhishTank dataset, which provides real-time, community-sourced phishing data, enhancing the real-world applicability of the model.
- Conducted a comparative performance analysis between GRU, AdaBoost traditional ML, and LSTM DL classifiers.
- Implemented an effective data preprocessing pipeline, including handling missing values, outlier removal, and feature encoding to improve data quality.
- Performed empirical hyperparameter tuning to optimize model parameters and achieve high detection accuracy with minimal overfitting.
- Demonstrated the superiority of GRU in phishing detection through extensive evaluation using measures, such as ROC-AUC, F1-score, recall, accuracy, and precision.

1.2. Justification and Novelty

Phishing attacks are getting more advanced, so it's tough for regular detection methods to cope with them. This work uses the sequential pattern of URL data to present a GRU-based phishing detection method, something that traditional ML models often overlook. Unlike LSTM, which is more complex and resource-intensive, GRU offers a computationally efficient yet equally effective alternative. The novelty lies in the combination of deep sequence Modeling, performance benchmarking, and lightweight architecture, making it suitable for real-time cybersecurity applications. The study fills a gap by providing a well-rounded evaluation of GRU in phishing detection, backed by comparative results and resource-aware deployment considerations.

1.3. Structure of Paper

The portions of this paper are listed below: Section 2 examines related research on detecting phishing attacks. Section 3 explains the proposed methodology, including data preprocessing and the GRU model. Section 4 displays experimental findings and performance evaluations. Section 5 ends the study and suggests the next research.

2. Literature Review

This section presents research on phishing detection systems that utilize diverse ML techniques. Table 1 summarizes the findings from this research.

Yazhmozhi and Janet (2019) propose a real-time solution for detecting phishing websites. To determine which method works best, it employs five distinct classification methods using NLP, as well as vector words, on two distinct feature sets. Using an accuracy of 97.99%, it was shown that the NLP-based RF method outperformed the other ML classification techniques, such as LR, SVM, DT, RF, and NB [6].

Wang *et al.* (2019) suggest a quick method for detecting phishing websites called PDRCNN, which classifies the original URL using a two-dimensional tensor en-

coded from it. The method automatically recognises important characters using a CNN and extracts global characteristics using a bidirectional LSTM network. With an AUC value of 99% and a detection accuracy of 97%, the PDRCNN outperforms state-of-the-art techniques. It just takes 0.4 ms to recognise objects [7].

Numerous researchers have researched online phishing detection systems; one of these researchers used data mining methods, although they only used one classification algorithm. As a result, Meta-algorithms should be incorporated to improve classification performance while developing various online phishing detection techniques. The UCI ML Repository's Web Phishing dataset is used in the process of testing, and adding the bagging method increased accuracy value by 97.1%, adding the boosting process increased it by 97.3%, and adding the stacking process increased it by 97.5%. Given the enhanced performance that follows, it is anticipated that the model might serve as a roadmap to improve the development of more phishing online detection tools [8].

If the domain holder is also a victim, or if an attacker has maliciously registered the domain. Here, they provide a brand-new ML domain classifier that uses only publicly accessible data to introduce characteristics based on a domain's online presence and history. A malicious domain feed and a phishing feed were sent for assessment by the Anti-Phishing Working Group. On harmful and hacked datasets from two different suppliers, their domain classification retains 88% as well as 92% accuracy, respectively, while achieving 94% accuracy on future harmful domains [9].

One of the most frequent and harmful cybercrime assaults is phishing. These attacks are designed to steal the data that people and organizations use to make purchases. Numerous clues may be found in the contents and browser-based data of phishing websites. In this research, 30 attributes, including Phishing Websites Data, Extreme Learning Machines (ELMs) from the UC Irvine ML Repository database, will be used to classify them. When compared to other ML techniques like SVM as well as Naïve Bayes, ELM showed the greatest accuracy of 95.34% after data analysis [10].

Hota, Shrivas and Hota (2018) focus on using the Remove-Replace Feature Selection Technique (RRFST) to build an ensemble ML-based model to identify phishing emails. By randomly choosing a feature and removing it if accuracy remains the same or improves, RRFST decreases features from the original feature space; if not, the feature is replaced with its original feature space. In order to create an effective classification model with a reduced feature subset acquired by RRFST, classifiers were created utilising two DT techniques, namely C4.5 and Classification and Regression Tree (CART), as well as an ensemble of these two. According to empirical findings, the suggested FST uses an ensemble of C4.5 and CART with just 11 characteristics to achieve an impressive 99.27% accuracy rate. Two current FSTs are also used to examine the research results of the proposed FST: Gain Ratio (GR) and Information Gain (IG) [11].

Subasi *et al.*'s (2017) research introduces an intelligent method for detecting phishing attacks. They adopted several data mining methods to label websites as either valid or emulation. Different approaches for classifying information were applied to improve intelligent systems that find phishing websites. F-measure metrics, area under the ROC curve, as well as classification accuracy, are used to evaluate data mining approaches. According to the results, with the highest accuracy of 97.36%, RF was the most successful. The random forest engine operates fast, and it is capable of phishing detection on several websites [12].

Table 1 compares phishing detection strategies utilising ML and DL. It shows methodology, datasets, and accuracy results, as well as future research areas such as DL integration, automated feature selection, adversarial training, and real-time detection.

Table 1. Summary on phishing attack detection for cybersecurity using deep learning approach.

Author(s)	Methodology	Dataset	Key Findings	Limitation & Future Work
Yazhmozhi and Janet (2019)	Five classification techniques that use natural language processing and word vector characteristics include DT, RF, NB, LR, and SVM	Custom dataset (not explicitly named)	NLP-based Random Forest features produced the greatest accuracy of 97.99%	Limited feature sets explored; future work could integrate DL or ensemble methods
Wang *et al.* (2019)	Proposed PDRCNN, combining Bidirectional LSTM for global feature extraction and CNN for key character identification from URL tensors	Custom-encoded URL dataset	Achieved 97% accuracy and 99% AUC; fast detection time (0.4 ms)	Model performance under varying attack types not evaluated; future work may explore robustness
Nugraha and Rahman (2019)	Meta-algorithm approach: Bagging, Boosting, Stacking applied to base classifiers	Web Phishing Dataset from UCI Repository	Accuracy improved to 97.5% with the stacking method	Focus only on ensemble methods; future work could integrate DL or adversarial training
Le Page and Jourdan (2019)	Domain classifier using internet presence and domain history features	APWG feeds (malicious and compromised domains)	Achieved 94% accuracy on future malicious domains	Only public domain info used; future models could incorporate real-time monitoring and DNS anomaly detection
Sönmez *et al.* (2018)	Comparing ELM to Naïve Bayes and SVM	Phishing Websites Data from UC Irvine ML Repository	ELM achieved the highest accuracy of 95.34%	Focused only on 30 features; future research could test with larger and more diverse feature sets

Continued

Hota, Shrivas and Hota (2018)	Developed an ensemble ML model using C4.5 + CART with RRFST for feature reduction in phishing email detection	Phishing Email Dataset	Achieved 99.27% accuracy using just 11 selected features with RRFST	Focused only on email-based phishing; could be extended to web-based phishing detection
Subasi *et al.* (2017)	Compared different classifiers; evaluated using Accuracy, ROC AUC, and F-measure	Dataset unspecified	At 97.36%, Random Forest had the highest accuracy	No discussion on feature engineering; future work could explore hybrid models and feature optimization

3. Methodology

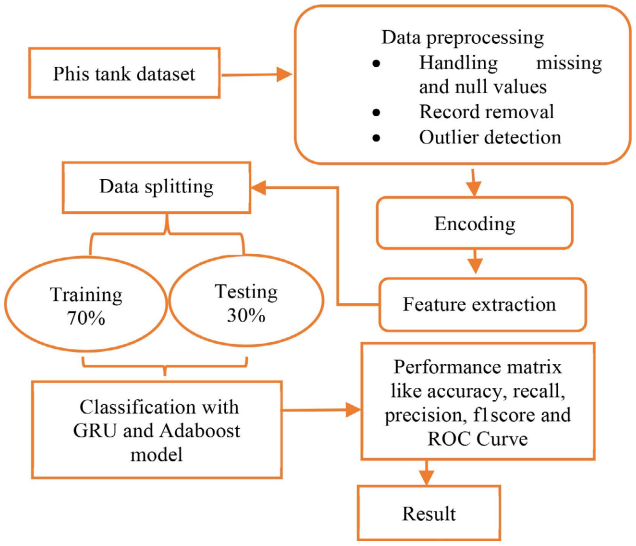

Figure 1. Flowchart of phishing attack detection for cybersecurity.

The strategy begins with acquiring the PhishTank dataset, which serves as the basis for developing and assessing models for phishing detection. To ensure high-quality input, the data undergoes several preprocessing steps, including handling missing or null values, removal of irrelevant records, and outlier detection. To facilitate efficient model validation, the preprocessed dataset is divided into training (70%) and testing (30%) subgroups. Both ML (AdaBoost) and DL (GRU, LSTM) models are used for categorisation. The GRU model, known for capturing sequential dependencies, is particularly suited for learning temporal patterns in URL structures, while LSTM and AdaBoost are used to strengthen comparative evaluation. In order to enhance model performance and avoid overfitting, the models are refined using empirical hyperparameter tuning, which involves carefully adjusting parameters including batch size, epochs, dropout rate, learning rate, number of GRU units, and embedding size. The efficacy of each model in identifying phishing attempts is evaluated by a thorough performance matrix that includes accuracy,

precision, recall, F1-score, and the ROC curve. Figure 1 illustrates the overall methodology adopted in this study.

The following sections provide a description of each step, which is also shown in the methodology and proposed flowchart.

3.1. Data Collection and Visualization

The PhishTank website, a constantly updated forum with phishing URLs that have been reported, provides the dataset for the phishing website URLs. PhishTank receives reports of phishing websites and has them retested by others to identify them as such. The PhishTank dataset, which comprises 20,000 URLs, is acquired and used. This dataset's visualization is shown below.

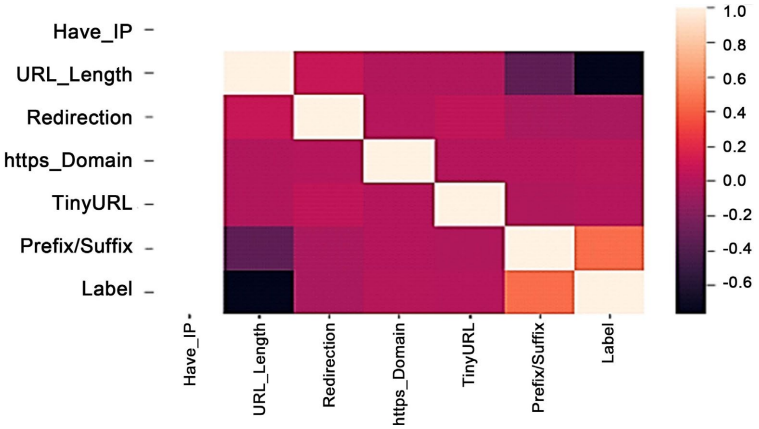

Figure 2. Heat map visualization of the features.

Figure 2 presents a heatmap illustrating the correlation matrix of various features used for detecting phishing URLs. In connection with the target variable Label, features like URL Length, HTTPs Domain, Tiny URL, Redirection, Have_IP, and Prefix/Suffix are examined. The colour gradient graphically depicts the direction and intensity of these correlations, ranging from bright peach (strong positive correlation) to dark purple (strong negative correlation). Interestingly, have_IP and Label show a high negative association, indicating that IP-containing URLs are more frequently associated with phishing. Conversely, the Prefix/Suffix feature shows a moderate positive correlation with the Label, indicating its potential significance in phishing detection.

Figure 3 displays a series of bar plots representing the distribution of values for various binary and categorical features in a phishing URL detection dataset. Each subplot corresponds to a specific feature Have_IP, URL Length, Redirection, HTTPs Domain, Tiny URL, Prefix/Suffix, and Label, highlighting the frequency of their values, typically 0 or 1. The plots reveal noticeable class imbalances; features like Have_IP, Redirection, and HTTPs Domain, for example, have a dominating count of 0, indicating that the majority of URLs do not display these attributes. Similarly, because the Label column mostly contains 0 s (indicating legitimate URLs), it could affect the model's accuracy unless made aware during training.

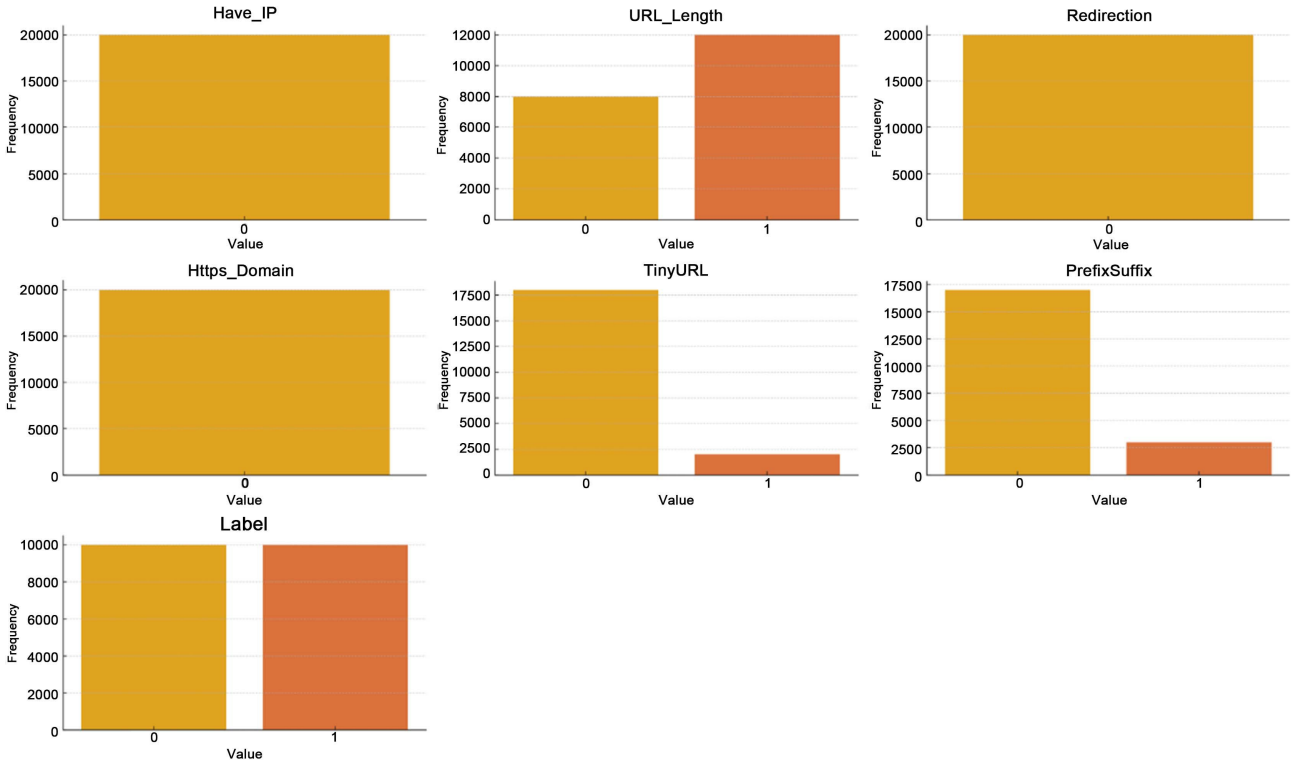

Figure 3. Features visualization.

3.2. Data Preprocessing

A process involves reviewing the data to see if there are missing or null values, removing any abnormal or missing data, using Z-score to handle outliers, sorting the information into features and labels and extracting features used to tell if URLs are suspicious. Preprocessing steps are introduced as follows:

- **Handling Missing and Null Value:** The dataset must be checked for missing or null elements before being fed into an ML model. Missing data in datasets may make it hard for the model to learn and give false predictions. Accuracy and resilience of the model depend on the dataset being full and consistent, which is ensured by identifying and resolving such items.
- **Record Removal:** In cases where missing or abnormal data entries are minimal, it might be more efficient to remove these records entirely. This approach is particularly useful when the removal does not compromise the dataset's integrity or representativeness.
- **Outlier Detection and Handling:** If outlier data points are not appropriately handled, they might skew model training by substantially differing from other observations. Several techniques can be used to detect and handle outliers.

3.3. Encoding

The text-to-sequence function was used to turn words into numbers, so that each word can be given a specific number as it comes up in the text. With this process, the text is changed into numbers that are suitable for DL algorithms to handle.

As it plans to talk about Glove in the latter part of this article, it chose to use texts_to_sequences since there are different encoding ways, like one-hot and TF-IDF.

3.4. Feature Extraction

Feature extraction is essential for phishing detection using the PhishTank dataset, as it helps identify characteristics typical of malicious URLs. This study extracts key features such as domain details (structure and age), URL characteristics (presence of IP, length, redirection, use of HTTPS in the hostname, short URLs, and prefix symbols like "-"), and HTML/content-based indicators (phishing-related keywords). It also includes link features (internal/external links), WHOIS data (domain age, registrar info), and SSL/TLS certificate attributes (issuer, validity). A DL model is trained and assessed using each characteristic, which is represented as a binary value (1 for phishing, 0 for authentic).

3.5. Data Splitting

The testing and training sets of the PhishTank dataset are separated in a 70:30 ratio. 70% of the data is used for model training, with 30% specifically designated for performance evaluation on unseen data.

3.6. Proposed GRU Model

The most popular variant of the GRU simplifies the gated architecture by using an update gate and a reset gate in place of three gates. While the updating gate decides how much of the old data needs to be stored in the current time step, the reset gate determines how to combine the most recent input data with the previously stored information. The following Equations (1)-(4) illustrate the fundamental calculation procedure of GRU. GRU has also produced good results in several well-known research fields and can save more training time and computer resources than GRU.

$$r_t = \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right) \tag{1}$$

$$z_t = \sigma\left(W_z x_t + U_z h_{t-1} + b_z\right) \tag{2}$$

$$\tilde{h}_t = \tanh\left(W_h x_t + U_h\left(r_t \odot h_{t-1}\right) + b_h\right) \tag{3}$$

$$h_t = \left(1 - z_t\right) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{4}$$

The output vectors of the update and reset gates at time step t are denoted by r_t and z_t, respectively. At time step t, h_t and $\tilde{h}_t$ represent the state & candidate state vectors, respectively. The weight matrices for the feed-forward as well as recurrent relationships are W_h, W_r, W_z, U_z, U_r, and U_h, whereas b_r, b_z, and b_h are the bias vectors. The weights are shared across different time steps. Elements are multiplied by the symbol $\odot$. As activation functions, the sigmoid function $\sigma(\cdot)$ and the hyperbolic tangent function $\tanh(\cdot)$ are employed. The following model hyperparameters are selected based on standard practices in

GRU-based sequence Modeling and prior literature to ensure optimal learning and generalization, as discussed in Table 2. They were fine-tuned through empirical testing to balance performance, training time, and overfitting control.

Table 2. GRU model hyperparameters.

Hyperparameter	Value
Embedding Size	128
GRU Units	64
GRU Layers	1
Dropout Rate	0.2
Optimizer	Adam
Learning Rate	0.001
Batch Size	64
Epochs	20

These hyperparameters were chosen based on common practices in sequence Modeling and prior research. For instance [13], Statistical Machine Translation's Learning Phrase Representations using RNN Encoder-Decoder showed that GRU with modest numbers of hidden units (e.g., 64 - 128) and embedding sizes between 100 and 300 perform well in text-based tasks. A dropout rate of 0.2 aids in minimising overfitting, while the Adam optimiser with a learning rate of 0.001 provides quick convergence. The epoch count and batch size were empirically adjusted to provide robust training.

3.7. Performance Matrix

In the experiment, the primary assessment indicators that it used were FPR and TPR, which are the industry standards for assessing anti-phishing tactics. The F1 measure, which aggregates FP and TP into a single summary statistic with equal weights, was also employed. To improve the ML models, it applied the concept of ROC curves. The Area under the Curve (AUC) measure of the ROC curve, a popular method for assessing the efficacy of binary classification, illustrates the trade-off between TP and FP. The performance metrics are listed below as [14]:

- **True Positive (TP):** The quantity of phishing emails that were correctly identified.
- **True Negative (TN):** The quantity of emails that were accurately recognised as authentic.
- **False Positive (FP):** The quantity of valid emails that were mistakenly labelled as phishing.
- **False Negative (FN):** The quantity of phoney emails that were mistakenly classified as authentic.

1) Accuracy

Accuracy is defined as the proportion of correctly classified cases to all occurrences,

as shown in Equation (5):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

2) Precision

The percentage of TP samples among those classified as positive samples is known as precision, as shown in Equation (6):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

3) Recall

Recall reflects the chance that a classifier successfully recognizes positive examples, as illustrated in Equation (7):

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

4) F1-Score

The F1-score is the recall as well as precision values' harmonic mean, as shown in Equation (8):

$$\text{F1} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \tag{8}$$

5) ROC

The effectiveness of binary classification methods is often assessed using the ROC curve. It explains how model performance fluctuates with changes in the threshold value. The ROC curve's vertical line represents the FPR, while the horizontal line represents the TPR.

4. Results and Discussion

Phishing experiments were carried out on powerful hardware to make sure the data was processed well and the model learned correctly. The machine had an RTX 2080 Ti GPU, 32 GB of DDR4 RAM, 11 GB of RAM, and a Core i7 CPU operating at 3.0 GHz. The studies with momentum sharing were conducted using Windows 10 Pro. The performance of the proposed models was evaluated using a variety of widely used measures, such as the ROC curve, loss, accuracy, precision, recall, and F1-score. Table 3 discusses the experimental findings. The following sections discuss the experimental results.

Table 3. Results for GRU model for phishing attacks on PhishTank dataset.

Matrix	GRU
Accuracy	98.01
Precision	98.67
Recall	98.41
F1-score	98.14

The GRU model's performance results for cybersecurity phishing attack detection using the PhishTank dataset are shown in Table 3. The model achieved 98.67% precision, 98.01% accuracy, 98.14% F1-score, and 98.41% recall. These results validate the GRU model's efficacy for the classification task and show its significant capacity to recognize phishing assaults with accuracy, exhibiting a high balance between precision and recall.

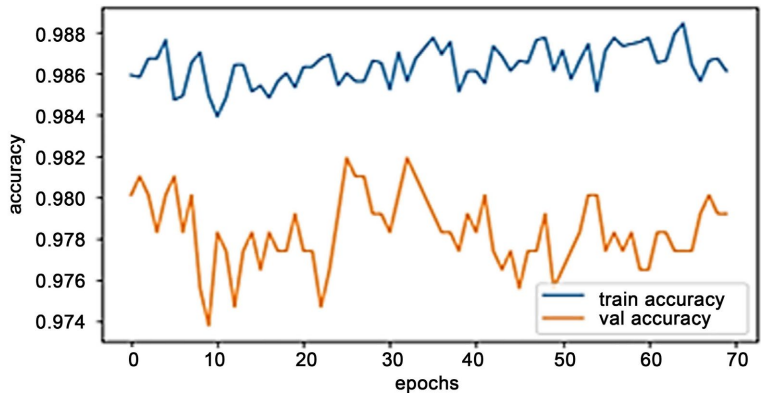

Figure 4. Training and validation accuracy graph of GRU model.

GRU model, a recurrent neural network variation that is often used in DL for sequential data processing, is shown in Figure 4 for both validation accuracy and training. Throughout 70 epochs, the training accuracy (depicted by the blue line) consistently remains high, fluctuating between approximately 0.984 and 0.988. In contrast, the validation accuracy (represented by the orange line) illustrates how well the model performs on hidden data, exhibiting greater variability, ranging from approximately 0.974 to 0.982.

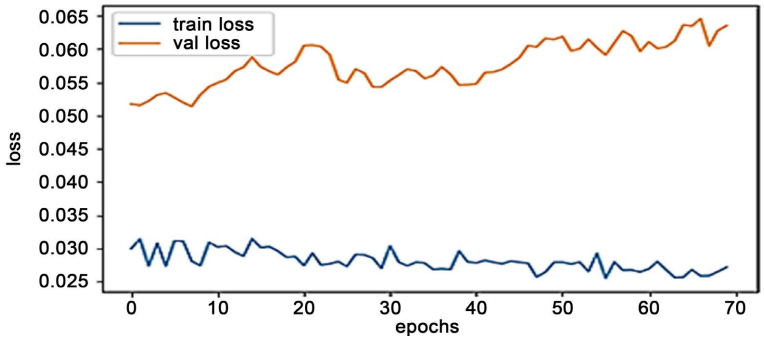

Figure 5. Training/Validation loss performance of GRU model.

As shown in Figure 5, a reduced validation loss indicates that the model more closely matches the learning procedure. The model is learning well and generalizing successfully in this instance, as shown by the reasonably low and near values of the validation loss (0.0537) and training loss (0.0318). On the other hand, a large validation loss suggests that the model is having trouble identifying patterns in the data.

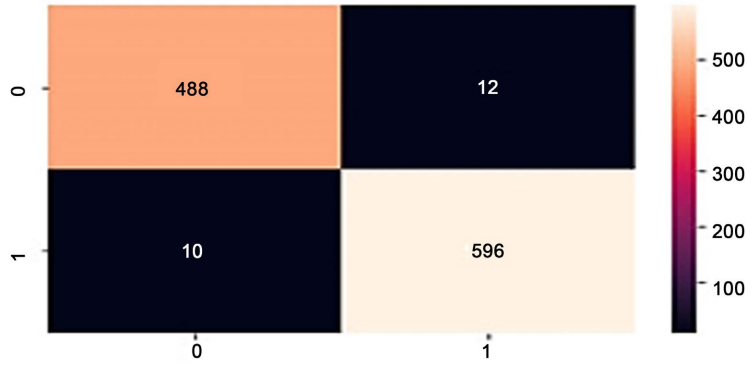

Figure 6. Confusion matrix of GRU model.

Figure 6 displays a confusion matrix that illustrates a model's capacity for categorization; 488 instances were accurately identified as class 0 TN, as well as 596 occurrences as class 1 TP. Ten class 1 cases were inadvertently given class 0 cases. While twelve cases of class 0 were inadvertently included in class 1. Overall, the GRU model performed well in this classification test, as seen by the comparatively large numbers of TP and TN relative to the FP and FN.

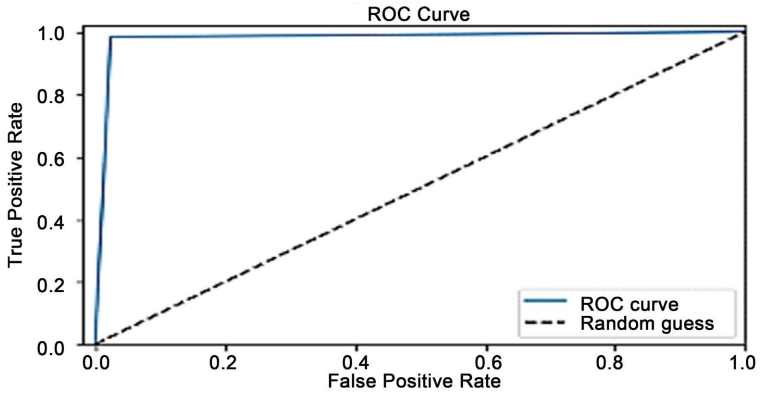

Figure 7. ROC curve of GRU model.

As the GRU model classifies different data, the True Positive Rate (TPR) in proportion to the False Positive Rate (FPR) is shown in ROC curve in Figure 7. Because of its almost perfect TPR and high rise in the top-left corner, the curve is particularly effective at discriminating. The area found under the ROC curve, or AUC, is around 1.0.

Table 4. Comparison between DL model's performance for phishing attack detection.

Matrix	GRU	AdaBoost [15]	LSTM [16]
Accuracy	98.01	93.24	95.15
Precision	98.67	90.8	92.01
Recall	98.41	96.3	87.2
F1-score	98.14	93.5	89.54

Table 4 presents a performance comparison between the proposed GRU model, AdaBoost, and LSTM, highlighting that GRU outperforms both traditional and deep learning baselines across all key evaluation metrics. GRU achieves the highest accuracy of 98.01%, compared to 95.15% for LSTM and 93.24% for AdaBoost, indicating superior overall performance. In terms of precision, GRU records 98.67%, surpassing LSTM (92.01%) and AdaBoost (90.8%), which means it makes fewer false positive predictions. GRU also achieves a recall of 98.41%, outperforming AdaBoost (96.3%) and LSTM (87.2%), showing it correctly identifies a larger portion of phishing instances. Lastly, its F1-score of 98.14% demonstrates a better balance between precision and recall, compared to 93.5% for AdaBoost and 89.54% for LSTM. These results clearly justify the selection of GRU, as it offers more accurate, consistent, and reliable phishing detection than both traditional and comparable deep learning models.

The result seeks to enhance performance by applying advanced DL, specifically GRU. From the preliminary studies, the GRU model will likely have higher recall, F1-score, accuracy and precision than traditional models such as Ad boost. The researchers believe that their approach, which addresses existing flaws and calibrates parameters, is likely to raise the bar in phishing attack detection and deliver improvements in how fast and correctly threats are found.

These results clearly justify the selection of the GRU model for phishing detection. GRU is particularly effective because it captures sequential dependencies in URL patterns, something traditional models like AdaBoost cannot do. Compared to LSTM, GRU provides similar or better performance with fewer parameters and faster training, making it both accurate and computationally efficient. This combination of high detection accuracy, low error rates, and reduced complexity makes GRU an ideal choice for real-world phishing detection systems.

5. Conclusions and Future Scope

A combined approach using ML and DL techniques was suggested and tested, allowing the end user to tell if an advertisement or website is genuine or phishing for phishing detection. This work presents a novel approach that uses a GRU DL model to identify phishing assaults. Using the PhishTank dataset, the GRU model accomplished an F1-score of 98.14%, 98.67% precision, 98.41% recall, and 98.01% accuracy in detecting phishing URLs. From the results, GRU was found to be better than traditional ML, especially in terms of performance using the evaluation metrics. The role of the GRU model is to find repeated URL features that mark phishing sites by their sequence. This analysis indicates that DL models have the potential to help tackle phishing detection, giving a strong cybersecurity answer for different needs. GRU has shown great results in this model, hinting at its potential for use in cybersecurity, especially to prevent phishing attacks. It adds to the research being done to boost online system security by applying advanced techniques from ML.

Work in the future aims to strengthen the GRU model by adding new datasets

and testing it with a range of phishing approaches. Optimizing hyperparameters, adding hybrid models and requiring continuous deployment for detecting phishing will be investigated. Furthermore, improving model interpretability through techniques like LIME and addressing adversarial robustness will be crucial to ensure long-term effectiveness and scalability in evolving cybersecurity environments.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Basnet, R., Mukkamala, S. and Sung, A.H. (2008) Detection of Phishing Attacks: A Machine Learning Approach. In: Prasad, B., Ed., *Soft Computing Applications in Industry*, Springer, 373-383. https://doi.org/10.1007/978-3-540-77465-5_19

[2] Hajgude, J. and Ragha, L. (2012) "Phish Mail Guard: Phishing Mail Detection Technique by Using Textual and URL Analysis". 2012 *World Congress on Information and Communication Technologies*, Trivandrum, 30 October 2012-2 November 2012, 297-302. https://doi.org/10.1109/wict.2012.6409092

[3] Chandra, J.V., Challa, D.N. and Pasupuleti, D.S.K. (2019) Machine Learning Framework to Analyze against Spear Phishing. *International Journal of Innovative Technology and Exploring Engineering*, **8**, 3605-3611. https://doi.org/10.35940/ijitee.l3802.1081219

[4] Kolluri, V. (2016) A Pioneering Approach to Forensic Insights: Utilization AI for Cyber-Security Incident Investigations. I*nternational Journal of Research and Analytical Reviews*, **3**, 919-922.

[5] Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A. and Marchetti, M. (2018) On the Effectiveness of Machine and Deep Learning for Cyber Security. 2018 10*th International Conference on Cyber Conflict* (*CyCon*), Tallinn, 29 May 2018-1 June 2018, 371-390. https://doi.org/10.23919/cycon.2018.8405026

[6] Yazhmozhi, V.M. and Janet, B. (2019) Natural Language Processing and Machine Learning Based Phishing Website Detection System. 2019 *Third International Conference on I-SMAC* (*IoT in Social, Mobile, Analytics and Cloud*) (*I-SMAC*), Palladam, 12-14 December 2019, 336-340. https://doi.org/10.1109/i-smac47947.2019.9032492

[7] Wang, W., Zhang, F., Luo, X. and Zhang, S. (2019) PDRCNN: Precise Phishing Detection with Recurrent Convolutional Neural Networks. *Security and Communication Networks*, **2019**, Article ID: 2595794. https://doi.org/10.1155/2019/2595794

[8] Nugraha, A.F. and Rahman, L. (2019) Meta-Algorithms for Improving Classification Performance in the Web-Phishing Detection Process. 2019 4*th International Conference on Information Technology, Information Systems and Electrical Engineering* (*ICITISEE*), Yogyakarta, 20-21 November 2019, 271-275. https://doi.org/10.1109/icitisee48480.2019.9003952

[9] Le Page, S. and Jourdan, G. (2019) Victim or Attacker? A Multi-Dataset Domain Classification of Phishing Attacks. 2019 17*th International Conference on Privacy, Security and Trust* (*PST*), Fredericton, 26-28 August 2019, 1-10. https://doi.org/10.1109/pst47121.2019.8949038

[10] Sönmez, Y., Tuncer, T., Gökal, H. and Avcı, E. (2018) Phishing Web Sites Features Classification Based on Extreme Learning Machine. 2018 6*th International Symposium on Digital Forensic and Security* (*ISDFS*), Antalya, 22-25 March 2018, 1-5.

https://doi.org/10.1109/isdfs.2018.8355342

[11] Hota, H.S., Shrivas, A.K. and Hota, R. (2018) An Ensemble Model for Detecting Phishing Attack with Proposed Remove-Replace Feature Selection Technique. *Procedia Computer Science*, **132**, 900-907. https://doi.org/10.1016/j.procs.2018.05.103

[12] Subasi, A., Molah, E., Almkallawi, F. and Chaudhery, T.J. (2017) Intelligent Phishing Website Detection Using Random Forest Classifier. 2017 *International Conference on Electrical and Computing Technologies and Applications* (*ICECTA*), Ras Al Khaimah, 21-23 November 2017, 1-5. https://doi.org/10.1109/icecta.2017.8252051

[13] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., *et al.* (2014) Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. *Proceedings of the* 2014 *Conference on Empirical Methods in Natural Language Processing* (*EMNLP*), Doha, 25-29 October 2014, 1724-1734. https://doi.org/10.3115/v1/D14-1179

[14] Xiang, G., Hong, J., Rose, C.P. and Cranor, L. (2011) CANTINA+: A Feature-Rich Machine Learning Framework for Detecting Phishing Web Sites. *ACM Transactions on Information and System Security*, **14**, 1-28. https://doi.org/10.1145/2019599.2019606

[15] Sahingoz, O.K., Buber, E., Demir, O. and Diri, B. (2019) Machine Learning Based Phishing Detection from URLs. *Expert Systems with Applications*, **117**, 345-357. https://doi.org/10.1016/j.eswa.2018.09.029

[16] Ren, F., Jiang, Z. and Liu, J. (2019) A Bi-Directional LSTM Model with Attention for Malicious URL Detection. 2019 *IEEE* 4*th Advanced Information Technology, Electronic and Automation Control Conference* (*IAEAC*), Chengdu, 20-22 December 2019, 300-305. https://doi.org/10.1109/iaeac47372.2019.8997947