International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 8, 2025
https://theaspd.com/index.php

A Comparative Study Of Artificial Intelligence-Based Models To Curb Prevalence Of Cybercrime

Anthony Ifeanyi Otuonye[1], Vivian Chinyere Mbamala[1], Innocent Harvey Ajunwa[1], Chika Norah John[1], Thaddeus Ogadimma Okonkwo[1], Chukwukere Okpani[1], Emmanuel Ubalatu[1], Tochi Chima Ewunonu[1], Edith Chidimma Otuonye[1], Mercy E. Benson-Emenike[1], Gloria Ngozi Ezeh[1], Ikechukwu Harrison Ezeh[1], Ifeoma Ibeneme-Sabinus[1], Chioma Ulunma Iroh[2]

[1]Federal University of Technology Owerri, Nigeria

[2]Abia State University Uturu, Okigwe, Nigeria

Abstract

Background: *Phishing remains a pervasive threat in the realm of cybersecurity, necessitating effective detection mechanisms to safeguard individuals and organizations from malicious attacks. However, deep learning can be used to effectively combat this evolving threat landscape following its success story in other fields of application.*

Objective: *For this study therefore, we seek to investigate the potentials of one-dimensional CNN-based phishing detection (1D-CNNPD) model and other existing deep learning techniques, with a view to addressing known challenges against accuracy of phishing email detection.*

Methods: *We carried out an experiment to assess the capability of the 1D-CNNPD models and developed an augmentation using a Long Short-Term Memory (LSTM) and a Gated Recurrent Unit (GRU) in order to give additional efficiency to its detection ability. Further experiment was carried out using two standard datasets as benchmark.*

Results: *It was discovered that appreciable model enhancement was achieved with additional layers to the base 1D-CNNPD model. The two benchmark datasets include the Spam Assassin, and the Phishing Corpus datasets. Performance results indicate that the 1D-CNNPD with Bi-GRU augmentations outperforms DeepAnti-PhishNet as well as other similar deep learning models, which were found to achieve better phishing email detection accuracy.*

Conclusion: *Our discovery that the Bi-GRU augmentation alone could achieve detection accuracy of 99.72% and an F1 score of 99.68% implies that the proposed enhancement is about 0.40% more efficient than the second-best performing model.*

Keywords : *Convolutional neural network (CNN), Deep learning, 1D-CNNPD, Phishing email, Cybersecurity*

INTRODUCTION

Detecting phishing emails has become an increasingly vital task in the realm of cybersecurity, given the rising sophistication of cyber threats and the significant risks posed to individuals, organizations, and even entire economies. Phishing attacks, which aim to deceive recipients into divulging sensitive information such as login credentials or financial details, continue to evolve in complexity and prevalence, making them a formidable challenge for both users and security systems alike. Machine learning methods has been adopted to address this escalating threat, as they present an opportunity to analyze extensive datasets and detect patterns indicative of such malicious behavior.

The effectiveness of phishing email detection systems heavily relies on the quality and quantity of data used for model training and evaluation. It is also worthy of note that the ever-changing landscape of phishing attacks, coupled with the immense influx of emails, pose challenges in curating a comprehensive and current dataset suitable for training machine learning models.

Traditional phishing email detection techniques rely heavily on human efforts to carry out feature analysis such as senders' identification, subject tittle, main content, and so on. Due to the complexity of recent attacks, however, there is need for continuous improvement of these detection techniques. Traditional phishing detection techniques often fail to correctly identify sophisticated phishing emails, leading to increasing need for a more intelligent AI-powered solution. Software Engineers can leverage on advanced Artificial Intelligence (AI) algorithms to analyze vast datasets and accurately detect phishing emails as well as respond quickly to threats. With good detection algorithm, phishing detection accuracy could be enhanced by correctly identifying even the most sophisticated anomaly and dynamically adapt to new and evolving phishing maneuvers.

[4] corroborates the fact that Machine Learning (ML) and Deep Learning (DL) approaches have proved to be effective in overcoming some of the limitations of traditional email-phishing detection approaches. The reason for this success story is not far-fetched: models based on ML and DL approaches can be trained to

learn phishing characteristics and patterns from large phishing datasets, during which process some of the features relating to phishing activities could be identified to aid the system.

This study therefore seeks to investigate the capabilities of the one-dimensional CNN-based phishing detection (1D-CNNPD) model and other existing deep learning techniques. We will carry out an experiment to assess the potentials of this models and further develop an augmentation using a Long Short-Term Memory (LSTM) and a Gated Recurrent Unit (GRU) to add greater efficiency to its detection ability.

LITERATURE REVIEW

2.1. Deep Learning Approaches to Phishing Detection

Deep learning approaches can easily extract useful features from emails, thus removing the need for labor-intensive feature extraction. This means that such approaches could capture a more comprehensive representation of email information inside the text content. For example, [16] made use of word embedding and Neural Bag-of-Ngrams with deep learning methods to detect phishing emails quite effectively, experimenting with a number of deep learning architectures, including CNN, RNN, LSTM, and MLP.

However, collecting relevant and representative data is a major determinant in developing effective and reliable detection mechanism for phishing email and in the fight to secure our cyberspace. The next section therefore presents a myriad of challenges that can significantly impact on the performance and reliability of data collection mechanisms for phishing detection.

3. METHODS

In this study, our major focus is to use deep learning techniques to correctly identify phishing email attacks. We are particularly interested in architectural design of deep learning models that most suitably fit email phishing detection.

In this section, we identify our datasets for this study as well as paint a comprehensive picture for its usage. We also identify the pre-processing steps, as well as a description of the trained models selected to perform our task.

3.1. Dataset Descriptions and Preprocessing

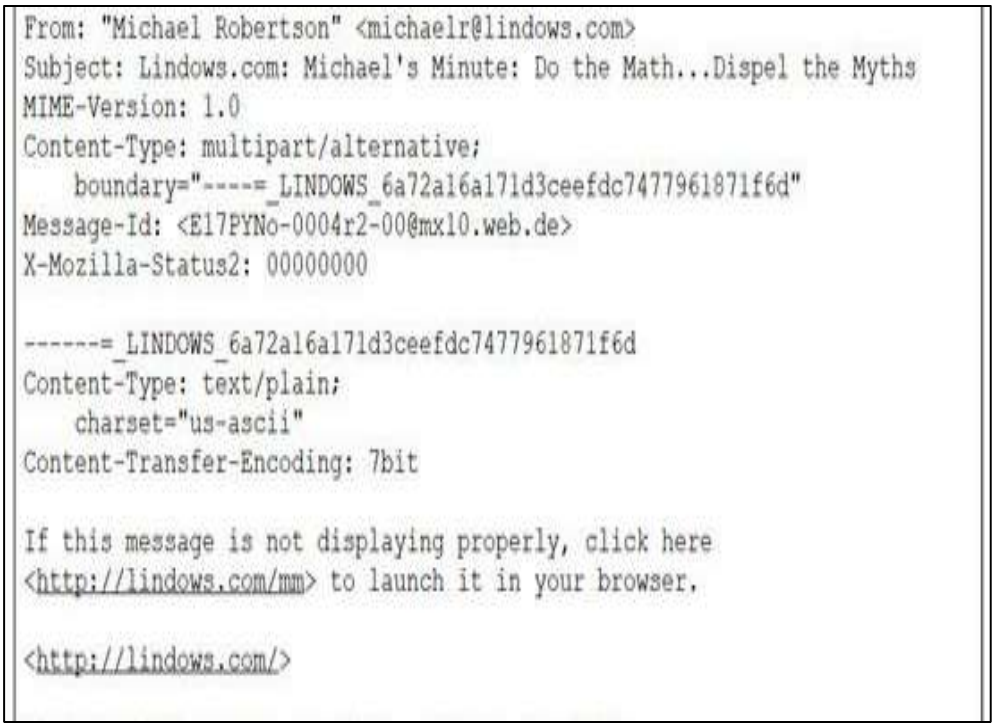
For the design of the proposed model, we made use of two publicly accessible datasets. They include:

- **The Spam Assassin dataset** [8]: This dataset is made up of both legitimate and spam emails directly gathered by the Spam Assassin algorithm. This dataset consists of approximately six thousand and forty-seven (6,047) sample emails, including one thousand eight hundred and ninety-seven (1,897) spam emails and four thousand one hundred and fifty (4150) legitimate emails.
- **The Phishing Corpus dataset** [9]: This public dataset is generally used for email phishing, made up of approximately seven thousand three hundred and fifteen (7,315) phishing emails that are gathered within a given time frame.

These datasets were chosen for this study because they are the most widely used. According to (19), more recent datasets may not necessarily be more advantageous than the older ones. Some of the more recent datasets are found to contain severe data imbalance, bringing a limited representation of more recent email phishing attacks.

For this study, a total of 6,428 emails were used to train our models, which was made up of a total of 4,150 legitimate emails and 2,278 phishing emails. The classification process therefore grouped all emails into two categories: phishing or legitimate.

All the emails identified in the datasets are in the email archive storage format, which consist of email header information, HTML tags, texts, encoded images and files, as well as URLs. A sample snippet in the dataset is shown in figure 1. At preprocessing stage however, all HTML tags, are eliminated, as well as all punctuations, unnecessary spaces, email addresses, IP addresses, images and attachments, while leaving out the main body of the mail.



```
From: "Michael Robertson" <michaelr@lindows.com>
Subject: Lindows.com: Michael's Minute: Do the Math...Dispel the Myths
MIME-Version: 1.0
Content-Type: multipart/alternative;
    boundary="----=_LINDOWS_6a72a16a171d3ceefdc7477961871f6d"
Message-Id: <E17PYNo-0004r2-00@mx10.web.de>
X-Mozilla-Status2: 00000000

------=_LINDOWS_6a72a16a171d3ceefdc7477961871f6d
Content-Type: text/plain;
    charset="us-ascii"
Content-Transfer-Encoding: 7bit

If this message is not displaying properly, click here
<http://lindows.com/mm> to launch it in your browser.

<http://lindows.com/>
```

Figure 1. Sample email snippet.

3.2. The Proposed Models

For the purpose of this study, we adopted the CNN-based models for email phishing attack detection for the fact that they are included in the best learning AI algorithms. CNN-based models have the rare capability of capturing hierarchical representations of data features, and consider both low-level and high-level abstractions. According to [18], CNN models have been used successfully in many cybersecurity problems, including phishing detection. In this study, we created a simple model design called 1D-CNNPD.

Next, we increased the model size in stages while calculating the performance of the model at each stage. Our interest is to discover the best model representation in order to develop a simple model (with short training times, relatively small parameters, and a fast inference capability) that can perform well in the detection of phishing attacks.

Finally, we introduced further improvement to our best 1D-CNNPD model by incorporating a Long Short-Term Memory (LSTM) and a Gated Recurrent Unit (GRU), together with their bidirectional variations. LSTMs and GRUs are specifically able to handle sequential data, allowing them to capture temporal dependencies effectively. Section 3.2.1. gives a more elaborate description of our models.

3.2.1. The 1D-CNNPD Model

Convolutional Neural Networks (CNNs) have the capability to extract essential features in tokens regardless of small positional variations with the various textual context. We choose to adopt the CNN model for this study because we are considering phishing emails as a document classification task. Each email message was regarded as a single document. The CNN was trained to take up a 1D input (email message) and output a document, classified as either phishing or legitimate.

Figure 2 illustrates our proposed architecture, made up of an embedding layer, convolution layer, a pooling layer, and a fully connected layer with sigmoid output.

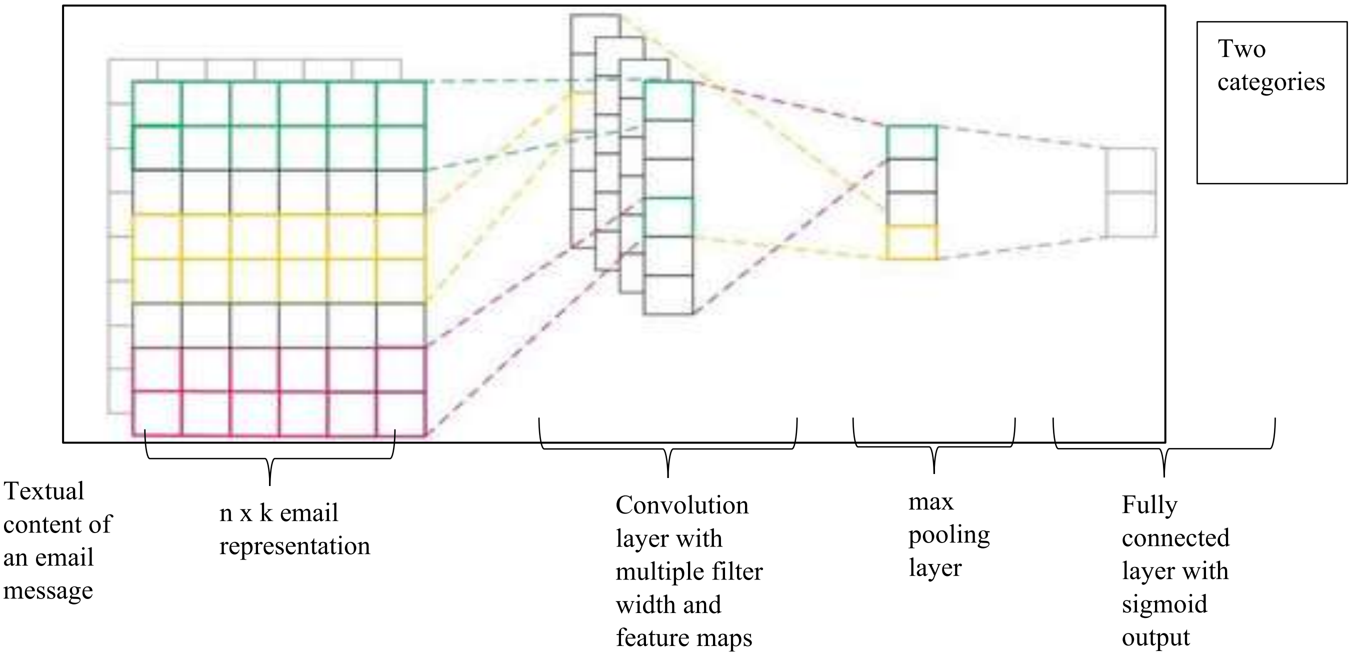

Two categories
Textual content of an email message
n x k email representation
Convolution layer with multiple filter width and feature maps
max pooling layer
Fully connected layer with sigmoid output

Figure 2. Our proposed CNN-based document classification model

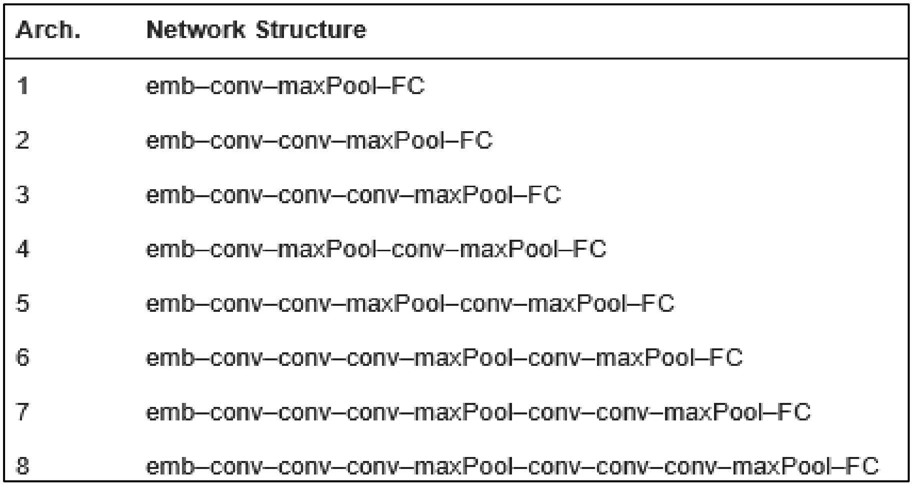
- **The Input Representation layer:** This layer of the models is made up of a list of tokens representing each email message.

- **The Embedding/ convolution layer:** This layer receives the list of tokens and transforms them into real-valued vectors represented in a lower dimensional space. We made use of the pretrained GloVe embeddings [20] of size 100, because GloVe has the capability to capture both global and local statistics of a corpus.

- **The 1D-CNN layer:** The embedding vector matrix is passed into a convolution layer, which extracts strong features that typically identify a phishing email. These features are then captured by the convolutional filters. In order to find the best hyperparameters possible, we varied the filter size (1, 5, and 10), the number of filters used (100, 300, and 600), and the activation function tested (ReLU, Tanh, and Linear). We made use of batch normalization (which we placed between the convolution layers and the pooling layer) in order to speed up the training process and avoid overfitting.
- **The Max-pooling layer:** At this stage, a max-pooling is applied to the output of the CNN layer. Max-pooling chooses the highest score among a matrix of features in the feature map (the pool size was set to 2). This layer primarily reduces the volume of output vectors, and by so doing, the output matrix is made smaller in size. The architectures are then tested with one or two pooling layers.

The different CNN architectures used in our experiment is shown in table 1.

Table 1. Different CNN architectures used in our experiment



Arch.	Network Structure
1	emb–conv–maxPool–FC
2	emb–conv–conv–maxPool–FC
3	emb–conv–conv–conv–maxPool–FC
4	emb–conv–maxPool–conv–maxPool–FC
5	emb–conv–conv–maxPool–conv–maxPool–FC
6	emb–conv–conv–conv–maxPool–conv–maxPool–FC
7	emb–conv–conv–conv–maxPool–conv–conv–maxPool–FC
8	emb–conv–conv–conv–maxPool–conv–conv–conv–maxPool–FC

- **The fully connected layer (FC layer):** The output of the max-pooling layer is a matrix that consists of 1D filters, which is then received by a fully connected (FC) layer. This layer employs the sigmoid function, as shown in Equation (1):

$h\theta(x)$ = 11 + $e-\theta Tx$ (1)

- **The output layer:** This layer finally produces the email's class as either phishing or legitimate, by the use of the FC layer's sigmoid activation function. According to [21], the sigmoid activation function is quite effective among other test functions when used in the last layer for text classification tasks.

3.2.2. Hyperparameter Tuning

We studied several combinations of hyperparameters with a view to selecting the best ones. The selection of the hyperparameters studied were based on [22], which suggests that the most important hyperparameters to consider in developing a CNN model for a document classification task should include the following: the word embedding technique, the kernel size, the number of filters, and the activation function used.

The kernel size was set to 1, 5, or 10. The filter value was set to 100, 300, or 600, while the activation functions are ReLU, Tanh, or Linear.

To tackle the problem of randomness, we set the seed parameter value to a fixed value. Finally, we applied the early stopping regularization approach in order to stop the training process if the performance begins to degrade. A summary of the hyperparameters is shown in table 2.

Table 2. Summary of hyperparameters.

Hyperparameter	**Value**
Seed	42
Loss	Binary-crossentropy
Number of layers	1, 2, 3, 4, 5, or 6
Kernel size	1, 5, or 10
Filters	100, 300, or 600
Activation function	ReLU, Tanh, or Linear
FC layer activation function	Sigmoid
Optimizer	Adam optimizer, learning rate of 0.001
Early stopping	monitor = 'val_loss', patience = 2
Batch size	32
Epochs	100

3.2.3. Our New/ Augmented 1D-CNNPD Model

Finally, we decided to augment the best performing 1D-CNNPD model using an LSTM and a GRU in order to give additional efficiency to its phishing detection ability.

To add this modification to the 1D-CNNPD model, we first of all discarded the fully connected layer and then added a ReLU layer with a Dropout. Next, we experimented with the following four augmentations: adding an LSTM layer, a Bi-LSTM layer, a GRU layer, and a Bi-GRU layer. The final model with the best augmentation (the Bi-GRU layer) is illustrated in figure 3.

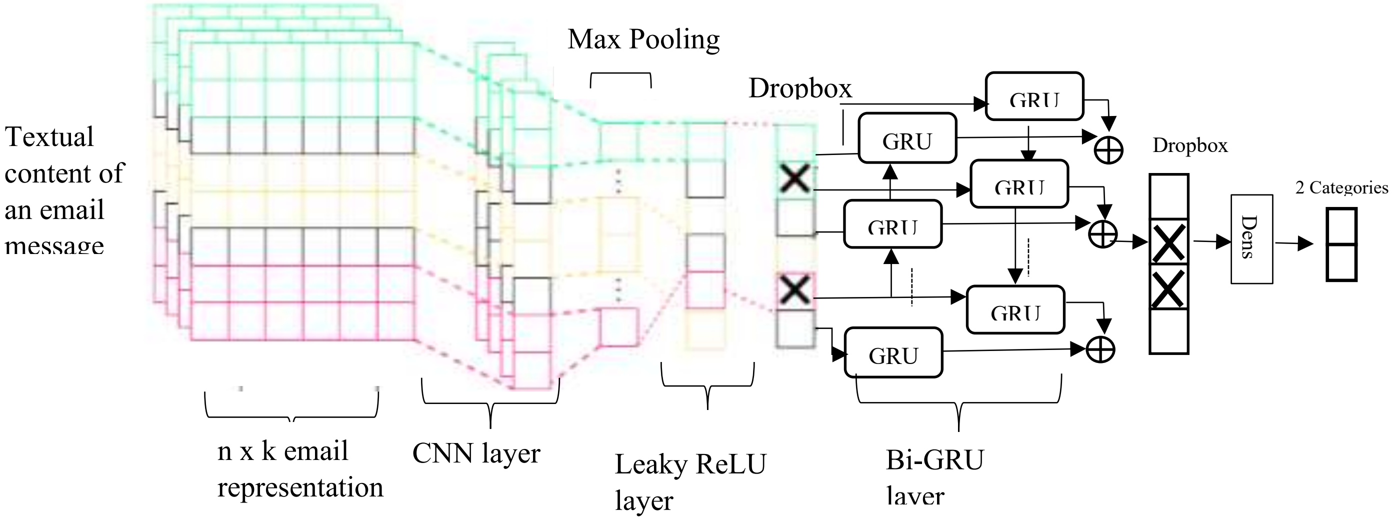


Figure 3. Augmented CNN with Bi-GRU and Leaky ReLU.

Model Description

a. The Leaky ReLU layer:

The Leaky ReLU layer is an activation function with a slight variation from the classic ReLU. Its output has a small gradient toward the negative input.

The equation (2) shows the Leaky ReLU; it could be a negative value instead of a zero value.

$Leaky{-}ReLU(x)$ = {$\alpha x x x$ < 0 x ≥ 0 (2)

where α is usually set as a small positive value, e.g., α = 1e−2

The motivation for using this Leaky ReLU layer derives from its ability to handle negative input datasets. The major difference between the Leaky ReLU and classical ReLU is shown in figure 4.

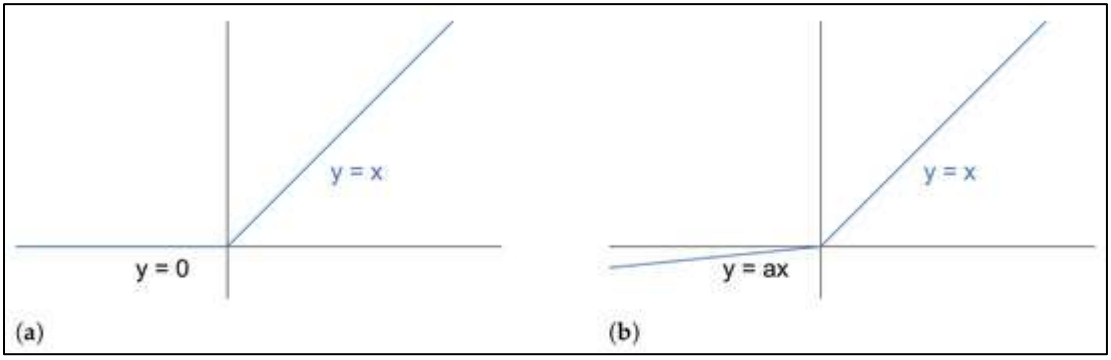


Figure 4. Main difference between Leaky ReLU and classical ReLU. (a) ReLU = $max(0,x)$; (b) Leaky ReLU = $max(\alpha x,x)$.

b. **Gated recurrent unit (GRU) layer**

The GRU layer preserves the original long short-time memory (LSTM) effect with a more direct structure, better accuracy, and fewer parameters. It is composed of two gates, one for updating and the other for resetting. The forward gate controls how much the output of the previous hidden layer affects the current layer (the more significant the value, the greater its effect). The reset gate controls how much of the previously hidden layer information is actually ignored.

The bidirectional gated recurrent unit (Bi-GRU) is divided into two unidirectional GRUs. The first is an output of the hidden layer state forward →, as shown in Equation (3), while the second ← is the backward hidden layer state, as shown in Equation (4).

$\mathbf{h}{\rightarrow}t$ = GRUFWD(xt,$\mathbf{h}t$−1) (3)

$\mathbf{h}{\leftarrow}t$ = GRUBWD(xt,$\mathbf{h}t$+1) (4)

where $h{\rightarrow}t$ is the forward GRU state, and $h{\leftarrow}t$ is the backward GRU state signifying the operation of concatenating two vectors.

There are two major reasons why we included the Bi-GRU rather than the Bi-LSTM in our model. The first is that the Bi-GRU has two gates, which is best suited for a small dataset of this nature. The second is that fitting fewer parameters for our small dataset saves time.

3.3. Performance Measures

Accuracy, Recall, Precision, and F1 score are the performance measures calculated for the evaluation of our email phishing detection models. The various performance measures are defined as follows:

i. **Accuracy**: The percentage level of correctly classified emails. Mathematically,

Accuracy = $TP + TN$

$TP + TN + FP + FN$ ——————— (5)

ii. **Recall**: The fraction of phishing emails correctly classified as phishing out of the total number of phishing emails.

Recall = TP

$TP + FN$ ——————— (6)

iii. **Precision**: The fraction of phishing emails correctly classified as phishing out of all the samples predicted as phishing emails.

Precision = TP

$TP + FP$ ——————— (7)

iv. **F1 score**: The harmonic mean of precision and recall.

$F-$Score = 2 × Precision × Recall
Precision + Recall ———————— (8)

Where,
TP (true positives) = Total number of emails that are correctly classified as phishing emails,
FP (false positives) = Total number of emails incorrectly classified as phishing emails,
FN (false positives) = Total number of emails incorrectly classified as phishing emails,
FN (false negatives) = Total number of emails incorrectly classified as non-phishing, and
TN (true negatives) = Total number of emails correctly classified as non-phishing.

3.4. Experimental Settings

Our new models were implemented using an open-source machine learning library called Tensorflow, which makes use of Keras. For the selection of the best performing model, we used the Talos hyperparameter tuning library (23). Experiments were carried out using GPUs running on the Google Colab environment, and for a more accurate evaluation of our models, we split the dataset into 70% for training and 30% for testing. The training set was further split into 70% for training and 30% for validation.

4. RESULTS

The purpose of our experiments was to evaluate the ability of CNN-based models to detect phishing emails. We experimented with a standalone CNN model, then the 1D-CNNPD, and finally the CNN model augmented with LSTM and a GRU, making use of two datasets as benchmark. The benchmark datasets are the Phishing Corpus [9], and the Spam Assassin [8] used for the training of our models. The performance results were compared with results of traditional machine learning techniques and with similar deep learning models.

4.1. Result of Hyperparameter Experiment

Results of the best hyperparameters for the various machine learning phishing detection architectures are shown in table 3. The models were trained on the training set and evaluated to select the best hyperparameters. These hyperparameters were chosen based on the best validation accuracy achieved during the tuning process.

Table 3. Results of hyperparameter optimization

1D-CNNPD	Activation	Kernel size	Filters
Arch.1	ReLU	1	300
Arch.2	Tanh	1	600
Arch.3	Linear	1	600
Arch.4	ReLU	5	100
Arch.5	Linear	1	100
Arch.6	Linear	1	300
Arch.7	Linear	5	100
Arch.8	Linear	5	300

4.2. Result of performance of the augmented 1D-CNNPD models compared with other deep learning techniques

Table 4 shows the performance measures of our new models and other similar models. It was observed that the augmentations in general, improved the performance of the base 1D-CNNPD model, with Bi-GRU yielding the best results. Augmentations with bidirectional LSTM and the GRU were also observed to improve performance over their unidirectional variations since they could equally capture long-range dependencies.

Table 4. Performance of augmented 1D-CNNPD models with other DL approaches

Deep Learning Algorithms	Dataset Phishing/Legitimate/Spam	Feature counts	Results	
Existing Models			**Accuracy**	**F1**
Deep-AntiPhisNet (16)	612/5088 train 4300 test	Body	98.2%	-
THEMIS (10)	4999/7781	Body and Header	99.24%	99.03%

Dual-layer CNN (17)	2664/5554	Body and content	99.28%	99.12%
CNN	14,950/3416	Body	96.34%	-
Augmented Models				
1D-CNNPD	1279/4150	Body and subject	98.87%	98.28%
1D-CNNPD with LSTM	1279/4150	Body and subject	99.23%	99.21%
1D-CNNPD with Bi-LSTM	1279/4150	Body and subject	99.32%	99.21%
1D-CNNPD with GRU	1279/4150	Body and subject	99.01%	99.62%
1D-CNNPD with Bi-GRU	1279/4150	Body and subject	99.72%	99.68%

4.3. DISCUSSION

A comparison of our results to DeepAnti-PhishNet [16] and other similar architectures shows that the extensions actually enhanced the accuracy and precision of email phishing detection, especially using the Bi-GRU. Our choice of these deep learning models for the comparison is based on studies that share a similar problem design to ours. Results indicate that the 1D-CNNPD with Bi-GRU augmentations performs better than both DeepAnti-PhishNet and THEMIS.

Furthermore, this study has shown that deep learning is indeed superior to traditional shallow learning in the problem area of email phishing detection. The intricate architectures of deep learning models allow easy extraction of features and contribute to a more effective learning process when compared to their shallow counterparts.

5. CONCLUSION

In this research, we have carried out investigation into existing deep learning models for phishing email detection, as well as developed an augmented 1D-CNNPD model. Our results indicate that the augmentation of a CNN for phishing detection with Bi-GRU could detect phishing emails more accurately, achieving a 99.72% accuracy, and an F1 score of 99.68%.

REFERENCES

[1] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Mirco (2018). On the Effectiveness of Machine Learning and Deep Learning Algorithms for Cyber Security. In Arc2018 10th International Conference on Cyber Conflict. Tallinn, Estonia., T. Minárik, R. Jakschis, and L. Lindström, Eds., Tallinn, pp. 371–390. doi: 10.23919/CYCON.2018.8405026.

[2] Verizon. 2022 Data Breach Investigations Report. 2022. Available online https://www.verizon.com/business/en-gb/resources/2022-data-breach-investigations-report-dbir.pdf (accessed on 13 December 2023).

[3] H. Al-Hamadi et al. (2019). A Novel Protocol for Security of Location Based Services in Multi-agent Systems. Wirel. Pers. Commun., vol. 108, pp. 1841–1868.

[4] Sahingoz, O.; Buber, E.; Demir, O.; Diri, B. Machine Learning Based Phishing Detection from URLs. Expert Syst. Appl. 2019, 117, 345–357. [Google Scholar]

[5] Macas, M.; Wu, C.; Fuertes, W. Adversarial examples: A survey of attacks and defenses in deep learning-enabled cybersecurity systems. Expert Syst. Appl. 2024, 238, 122223. [Google Scholar] [CrossRef]

[6] Bagui, S.; Nandi, D.; Bagui, S.; White, R.J. Machine learning and deep learning for phishing email classification using one-hot encoding. J. Comput. Sci. 2021, 17, 610– 623. [Google Scholar] [CrossRef]

[7] Glenn, A.; LaCasse, P.; Cox, B. Emotion classification of Indonesian Tweets using Bidirectional LSTM. Neural Comput. Appl. 2023, 35, 9567–9578. [Google Scholar]

[8] Mason, J. The Apache SpamAssassin Public Corpus. 2005. Available online:
https://spamassassin.apache.org/old/publiccorpus/20050311_spam_2.tar.bz2

[9] Nazario. Phishing Corpus 2015. 2015. Available online:
https://academictorrents.com/details/a77cda9a9d89a60dbdfbe581adf6e2df9197995a

[10] N. Majumdar, S. Shukla, and A. Bhatnagar (2019). Survey on applications of internet of things using machine learning. Proc. 9th Int. Conf. Cloud Comput. Data Sci. Eng. Conflu. 2019, no. January, pp. 562–566. doi: 10.1109/CONFLUENCE.2019.8776951.

[11] M. D. Behera et al. (2021). Species-level classification and mapping of a mangrove forest using random forest—utilisation of aviris-ng and sentinel data. Remote Sens., vol. 13, no. 11. doi: 10.3390/rs13112027.

[12] V. S. Mohan, J. R. Naveen, R. Vinayakumar, and K. P. Soman (2018). A.R.E.S: Automatic rogue email spotter. In R. Verma, A. Das (eds.): Proceedings of the 1st AntiPhishing Shared Pilot at 4th ACM International Workshop on Security and Privacy Analytics (IWSPA 2018), Tempe, Arizona, USA. pp. 57–63. [Online]. Available: http://ceur-ws.org .

[13] A. Jain et al. (2020). Overview and Importance of Data Quality for Machine Learning Tasks. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York; NY; United States. pp. 3561–3562. doi: 10.1145/3394486.3406477.

[14] P. Finn and M. Jakobsson (2007). Designing ethical phishing experiments," IEEE Technol. Soc. Mag., vol. 26, no. 1, pp. 46–58. doi: 10.1109/MTAS.2007.335565.

[15] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat (2021). A comprehensive survey of AI-enabled phishing attacks detection techniques. Telecommun. Syst., vol. 76, no. 1, pp. 139–154. doi: 10.1007/s11235-020-00733-2.

[16] Vinayakumar, R.; Barathi Ganesh, H.B.; Anand Kumar, M.; Soman, K.P. DeepAnti- PhishNet: Applying deep neural networks for phishing email detection. In Proceedings of the 1st AntiPhishing Shared Pilot at 4th ACM International Workshop on Security and Privacy Analytics (IWSPA 2018), Tempe, AZ, USA, 21 March 2018.

[17] Doshi, J.; Parmar, K.; Sanghavi, R.; Shekokar, N. A comprehensive dual-layer architecture for phishing and spam email detection. Comput. Secur. 2023, 133, 103378. [Google Scholar] [CrossRef]

[18] Alshingiti, Z.; Alaqel, R.; Al-Muhtadi, J.; Haq, Q.E.U.; Saleem, K.; Faheem, M.H. A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM- CNN. Electronics 2023, 12, 232. [Google Scholar] [CrossRef]

[19] Salloum, S.; Gaber, T.; Vadera, S.; Shaalan, K. A systematic literature review on phishing email detection using natural language processing techniques. IEEE Access 2022, 10, 65703–65727. [Google Scholar] [CrossRef]

[20] Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543. [Google Scholar]