Journal of Information Systems Engineering and Management
2025, 10(27s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

AMPDF: A Hybrid Deep Learning Framework for Multi-Modal Phishing Detection in Cybersecurity

Amna Kadhim Ali[1], Arkan A. Ghaib[2], Mustafa Al-atbee[3], Zaid Ameen Abduljabbar[4,5]

[1]*College of Veterinary Medicine, University of Basrah, Basrah, 61004, Iraq,* amna.kadhim@uobasrah.edu.iq

[2]*Department of Information Technology, Management Technical College, Southern Technical University, Basrah, 61004, Iraq.* arkan.ghaib@stu.edu.iq

[3]*Department of Computer Science, Shatt Al-Arab University College, Basra, Iraq.* mustafa.subhi@sa-uc.edu.iq

[4]*Department of Computer Science, College of Education for Pure Sciences, University of Basrah, Basrah 61004, Iraq,* zaid.ameen@uobasrah.edu.iq

[5] *Department of Business Management, Al-Imam University College, Balad 34011, Iraq*

ARTICLE INFO

Received: 29 Dec 2024

Revised: 17 Feb 2025

Accepted: 27 Feb 2025

ABSTRACT

Phishing is a type of cyberattack where attackers deceive users into revealing sensitive information via fake emails or fake websites. With the rise of online banking and social networking, they have been taking advantage of user vulnerabilities instead of network flaws, and phishing attacks have become more advanced. Such attacks generally involve emails containing harmful links that lead victims to fake sites that can swipe personal data. Classic anti-phishing tools heavily depend on blocklists and allowlists which make them ineffective against novel attacks leading towards high false positive rates. To overcome this, we introduce the Adaptive Multi-Modal Phishing Detection Framework (AMPDF). This data-driven hybrid model identifies phishing via analyzing three datasets: URL data, page content as well as traffic data using AI techniques. For these features, AMPDM Uses a CNN to extract the URL pattern then feeds it into a Dense layer with a LeakyReLU activation to analyze the page and finally another Dense layer for traffic behavior. These extractors integrate into a Fusion Layer followed by a Dense Layer with Dropout to avoid overfitting and a final classification layer splits phishing from legitimate instances. The model was evaluated using precision, recall, and accuracy metrics, with 98% accuracy and a test loss of 0.0708 on the test dataset. The experiments justify AMPDF as a significant and non-intrusive detection model against the traditional methods that are popular.

Keywords: Deep Learning, Cybersecurity, Phishing Detection, Multi-Modal Analysis, Hybrid Model

INTRODUCTION

The world has become more dependent on the Internet and digital technologies in all aspects of life, particularly in social media, online shopping, banking and financial services in the digital age [1]. With this dependence, many security threats have been realized already by exploiting technical as well as human weaknesses for financial advantage. One of the most prevalent, sophisticated, and evolving cyber threats you can read about is also phishing [2]. Phishing is an attack mechanism used by the attacker to impersonate legitimate users or organizations using fake messages that seem to come from a trusted source [3], [4]. The main purpose is to obtain critical information like login credentials, credit card details, or unauthorized access to crucial systems. These types of attacks involve sophisticated social engineering and psychological manipulation techniques used to deceive and exploit the victims [5]. In fact, the study of phishing is becoming more and more important as attacks evolve and there is a wider range of targets [6]. Demand the need for strong detection and prevention mechanisms. This has led to the evolution of contemporary phishing detection methods that leverage AI and ML models capable of examining sophisticated structures and detecting fraudulent events with strong accuracy and efficiency [7], [8], [9]. Even with the advances in security technologies, attackers keep evolving by using more advanced strategies, including zero-day phishing attacks, deep fake-based impersonation, and automated phishing campaigns [10]. As these strategies evolve, standard security practices such as rule-based detection and blocklisting become less and less effective. Hence, advanced hybrid models based on the combination of heuristic analysis, natural language processing (NLP), and deep learning approaches have emerged to mitigate phishing [11], [12]. Encouraging popular and corporate knowledge about it is also one of the most important ways to defend against cybersecurity threats. In fact, according to [13]. The main reason phishing attacks are successful is the human factor, which is when users open malicious links or download infected attachments. Systematic security awareness training programs, multi-factor authentication (MFA), and real-time phishing alert systems are examples of countermeasures that can reduce the risks of being a target [14]. Given the evolving nature of phishing, an effective process to combat this growing cyber threat lies in a layered defense strategy that combines technological innovations, people awareness, and proactive threat intelligence [15], [16].

This paper provides a comprehensive survey of the various factors involved in phishing, analyzes the attacker's techniques reviews the anti-phishing techniques and approaches and suggests the use of multi-modal data processing in a hybrid model to accurately and effectively detect phishing. Our research aims to push the boundaries of digital security developing cutting-edge timely tools to recognize and manage phishing risk. By presenting these findings we aim to inform and empower cybersecurity professionals, researchers and organizations interested in digital security. The study will provide insights into the evolving nature of phishing attacks, which are becoming increasingly reliant on sophisticated social engineering techniques as well as automation and deception, by systematically assessing different tactics used by attackers. The survey focuses on the limitations of traditional techniques and highlights real-time or modern detection techniques like machine learning models, AI based detection systems, and hybrid detection stations. Instead, this model combines several data such as URL, page content, and traffic to improve the detection approach. Using both decompilers and nontraditional classifiers, the model can improve the discovery of phishing attempts including those that would fly under the radar of conventional detection methods. Finally, we hope this study will lead us towards innovative, practical, and scalable solutions that support integrating the mentioned solutions into existing security infrastructures to decrease the potential danger of phishing attacks. Thus, the results of this research will contribute to creating a safer digital space and instilling a more proactive mindset in terms of phishing defense.

Here is what we contribute to this line of research:

1. Feature Engineering: Available three data types URLs, page content, and traffic behavior offered the generic features of detecting phishing status.
2. Data Generator: The data was not loaded fully into memory; a data generator was used to optimize training efficiency.
3. Our Solution: The hybrid learning architecture CNN + dense layers: The proposed model was a hybrid learning architecture made up of CNN and dense layers, which added flexibility to accommodate with advanced phishing techniques.
4. Scalability: The model is lightweight, using batch training with a low learning rate, making execution time a minimal concern.

RELATED WORK

There are numerous previous academic studies that describe what happens with phishing sites. The proposed approach is rooted in concepts from previous work and addresses its shortcomings, and aims to leverage the associated areas for improvement. Recent research [17] proposed a neural network-based phishing detection system that works on URL analysis. Two well-known models, ANN and DNN, were used with 37,175 phishing URLs and 36,400 legitimate URLs combined into a dataset and trained using 27 different features extracted from each URL. The DNN model provided an accuracy of 96%, which is higher. However, there are some limitations in the system, including dependency on static URL features, which might not be future-proof against evolving phishing techniques and greater execution time since some external URLs require services such as Alexa ranking. Moreover, they mentioned the scalable nature of their study yet did not compare their work with more sophisticated models, like CNNs or RNNs. Their work [18] proposes our approach for the detection of phishing URLs using a CNN that uses a one-hot character-level encoding so that the approach does not need handcrafted features or an external database. This is a lightweight and efficient model, making it mobile deployable and able to detect zero-day attacks. With public datasets, the accuracy of CNN surpasses that of LSTM based approaches without sacrificing computational efficiency. Nonetheless, there are some limitations, such as concerns in generalizability, potential challenges in adaptation to changing phishing strategies, scalability issues, no multi-modal analysis, and the necessity for continuous retraining to retain effectiveness. Dynamic Phishing Safeguard

System DPSS and deep learning-based intrusion detection system IDS based phishing detection are discussed in this research [19].

Using APNA and APBA, DPSS achieves 97.82% accuracy using 30 features of the URL, also allows real-time security like alerts and block the website. With a hybrid CNN-RNN model, the IDS extracts phishing indicators from the network traffic and website characteristics. The DPSS is limited in scalability and adaptability, while the IDS necessitates data and resource-intensive retraining, is restricted in interpretability, and is subject to adversarial attacks. By analyzing both network traffic and website features, the research [20] proposed a deep learning-based IDS specifically for phishing detection, applying CNNs and RNNs. Leveraging a hybrid approach to feature analysis, the system would be able to detect phishing in real-time by spotting complex scanning patterns. Nevertheless, this approach has some limitations, which include the ongoing process of retraining, limited interpretability of deep learning models, dataset limitations (KDD-CUP99) for modern phishing techniques, and vulnerability to adversarial attacks. The study [21] proposes an RNN-based framework for the phishing detection system, which does not require any manual feature selection; it targets its focus on phishing emails by applying NLP to analyze textual structures. The model is trained with LSTM layers allowing it to learn language rules and achieve high precision and recall as well as adapt to new types of phishing attacks making it suitable for real-time use. It does not handle non-text based phishing methods (e.g., all websites and multimedia-based attacks) and may not be effective against obfuscation methods. A real-time phishing detection model for edge devices is proposed in [22], which discusses the use of quantized machine learning models suitable for low-energy devices. This approach improves privacy and limits dependency on external servers by analyzing data at the source. It uses quantized lightweight neural networks with ReLU activation and binary cross-entropy loss, which are fast and secure. It does not use advanced deep learning techniques (such as CNNs or RNNs), which could increase detection accuracy. Its adaptability to changing phishing tactics and effectiveness against a range of attack strategies remains unconvincing. The system Phish Haven focuses on identifying phishing URLs generated by AI and Human [23] which the study presents. It applies lexical feature analysis, URL HTML encoding and novel URL Hit for real-time classification, even for tiny URLs. The system utilizes ensemble machine learning with multi-threading to enhance speed and accuracy and provides unbiased voting to minimize false classifications. It is particularly gratifying that so much work by existing systems is devoted to finding very carefully hidden human-domain phishing. Still, this problem has not attracted much interest, so it is terrific to see that we are able to detect AI-generated URLs as phishing vectors. It is based on lexical features, the same as Deep Phish that causes it to be less effective against upcoming stronger AI systems.

In the research [24], the authors proposed a phishing detection model that brings the ability to learn features (or patterns) directly from the URL using a character-level Convolutional Neural Network (CNN) without using any manual feature engineering or third-party services. The model embeds the URL characters using one-hot encoding to generate a fixed length sequence, and then it applies convolutional and pooling layers to detect phishing patterns. Through experimentation on different datasets, the CNN model has shown to be better than traditional machine learning models, such as Logistic Regression and Random Forest, with higher accuracy in the detection of phishing URLs. The existing shortfalls are defined as the model's high training time, which means it may not be a viable solution for time-sensitive applications, that it does not verify URL activity which may affect accuracy in real-world settings, and that the use of short or misleading URLs may cause misclassification. Disadvantages Gate Ensemble The model can struggle against complex phishing attacks, offering unique insights into the nature of such attacks; however, it may be limited in its ability to fully exploit semantic information or sophisticated URL structures, as it operates on a character level. Phishing URL detection is discussed in [25] and it proposes a hybrid deep learning model developed from Deep Neural Network (DNN) and Long Short-Term Memory (LSTM) architectures. This model combines the capabilities of DNN for understanding high-level NLP features and LSTM for recognizing sequential patterns in URL characters. It is based on two datasets, one of which is a newly developed dataset, and the study compares the hybrid model to classical machine learning and standalone deep learning approaches. Due to the different size of the feature set, SVM, RF, and DNN, as mentioned previously, while detected phishing URLs in both feature size and detection accuracy as compared to DNN-BiLSTM. However, the study has 5 gaps: (1) The model has limited generalizability, given that it has only been validated on two datasets, (2) There is no exploration of more complex forms of feature selection, (3) Due to the merging of DNN and BiLSTM, the computational cost is quite high, (4) The model has limited ability to adapt to new and evolving phishing techniques in real time, (5) The study does not compare against more advanced approaches, such as transformer-based models.

PROPOSED METHOD

The approach suggested an enhanced phishing detection using a hybrid deep learning model. Multiple data sources—URL data, page content, and traffic data—are analyzed. The outputs of these sources are combined in a single model to increase detection accuracy and minimize errors. The overall structure of the proposed framework is illustrated in Figure 1.

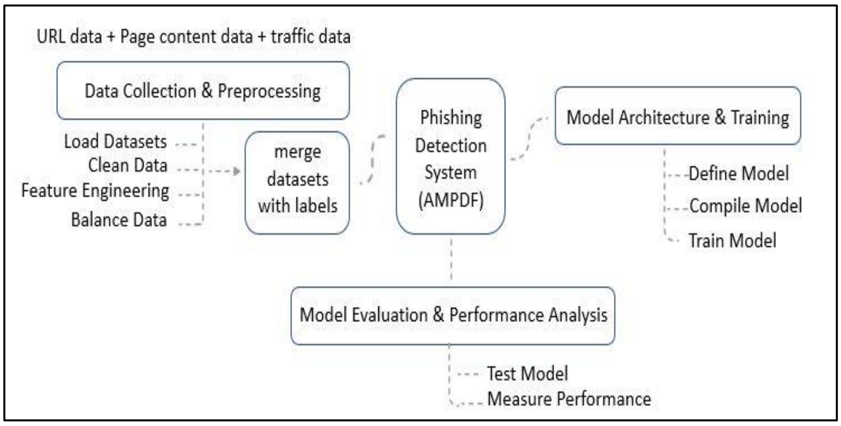
URL data + Page content data + traffic data
Data Collection & Preprocessing
Load Datasets
Clean Data
Feature Engineering
Balance Data
merge datasets with labels
Phishing Detection System (AMPDF)
Model Architecture & Training
Define Model
Compile Model
Train Model
Model Evaluation & Performance Analysis
Test Model
Measure Performance

Figure 1: The proposed method

1. Dataset

The three datasets used in this research were obtained from Kaggle. The first dataset consists of 822,000 URL links classified as either trusted or phishing (https://www.kaggle.com/datasets/harisudhan411/phishing-and-legitimate-urls).The second dataset comprises 18,000 emails categorized as either safe or phishing(https://www.kaggle.com/datasets/subhajournal/phishingemails/data).The third dataset contains traffic data with 2,000,000 rows and 79 features (https://www.kaggle.com/datasets/sweety18/cicids2017-full-dataset), classified as either **BENIGN** or one of the following attack types: **DoS Hulk, PortScan, DDoS, DoS GoldenEye, DoS Slowloris, DoS Slowhttptest, Bot, Infiltration, and Heartbleed**, which represent unsafe categories.

Based on the data types of the datasets, three different pre-processing techniques were applied to them before being merged together into one unified dataset suitable for hybrid_model. The below describes the steps that the data took before arriving in the final dataset:

1-1 URL dataset

In the data processing phase, we began with cleaning the data -- removing irrelevant columns, and dealing with missing values, duplicates, outliers, etc. After that, we extracted features using URL properties like the previously mentioned length of the URL or number of dots, domain, and subdomain. Then we converted categorical data to numerical data, balanced out the classes (We're going to work with classification algorithms so having equal classes is crucial), normalized our data -- series vs data frame and many other steps. Figure 2 represents the different URL data processing stages.

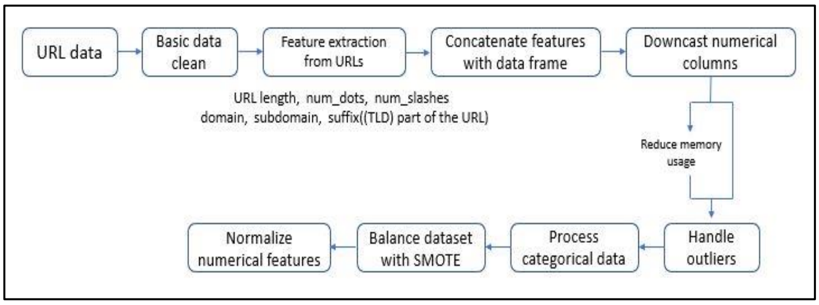
URL data
Basic data clean
Feature extraction from URLs
Concatenate features with data frame
Downcast numerical columns
URL length, num_dots, num_slashes
domain, subdomain, suffix((TLD) part of the URL)
Reduce memory usage
Handle outliers
Process categorical data
Balance dataset with SMOTE
Normalize numerical features

Figure 2: The stages of URL data preprocessing

1.2 Page dataset

The following are the stages in which the email data is processed with the goal of cleaning the data and transforming it into a format that will work in a modeling environment while improving the quality of the specified features. Data is cleaned by dropping irrelevant columns and cleaning missing and duplicate values for reliability. Later, it will remove numbers and special chars, and convert the text into lower case for uniformity to process the email texts. After that, categorical values in the dataset are encoded (Converting any textual feature such as domain names or categories to numerical representations) using a label encoder to adapt them to machine learning algorithms. It then further transforms these into a numerical "bag-of-words" representation given by Count Vectorizer. This technique produces numerical vectors according to the frequency of certain words used in the texts while the number of features is determined in relation to selecting the most frequently used words in the dataset to attenuate the value of less important features. To overcome class imbalance, we used Synthetic Minority Over-sampling Technique (SMOTE) to produce synthetic samples of the minority class so that the model will not be biased toward the majority class. To improve the performance of the model, low variance features were subsequently dropped and important features were selected using Random Forest Classifier. We also determined the feature importance based on its involvement in the decision-making process and removed features with low importance, thus reducing the dimensionality and enhancing the efficiency and accuracy of the model. For example, the page data processing process is illustrated in the following figure (Figure 3).

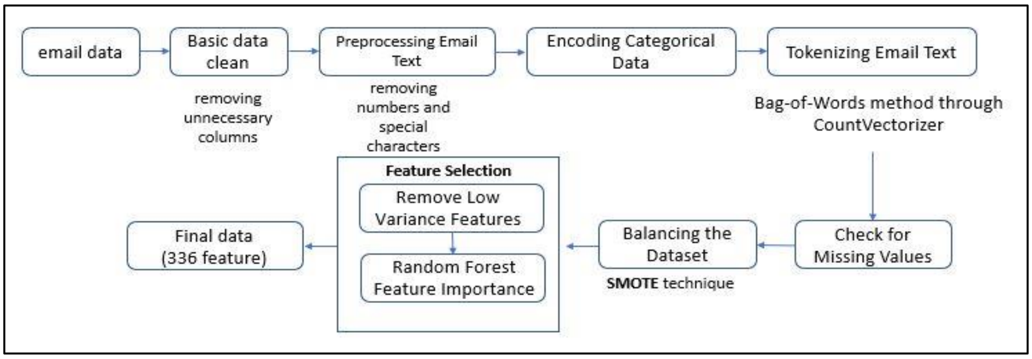
email data
Basic data clean
removing unnecessary columns
Preprocessing Email Text
removing numbers and special characters
Encoding Categorical Data
Tokenizing Email Text
Bag-of-Words method through CountVectorizer
Check for Missing Values
Balancing the Dataset
SMOTE technique
Feature Selection
Remove Low Variance Features
Random Forest Feature Importance
Final data (336 feature)

Figure 3: The stages of page data preprocessing

1.3 Traffic data

The traffic data contains normal traffic data and nine types of unsafe traffic data. In the malicious data, the first step is to merge all phishing data into one category, thus creating two final categories: benign data and unsafe data. Given that the benign category is larger, we need to downsize it next to ensure the size of this category matches the size of the unsafe category. The data are then merged and shuffled into the two categories after balancing.

The next step in data preprocessing is to identify the text columns and convert them to numerical values, using Label Encoder to transform categorical values to integers; this is followed by using Standard Scaler to standardize the numerical values to ensure that all features fall within a homogeneous range.

Finally, PCA dimensionality reduction is executed to keep 95% of the variance in a bid to reduce the number of features. The diagram in Figure 4 depicts the steps in processing traffic data.

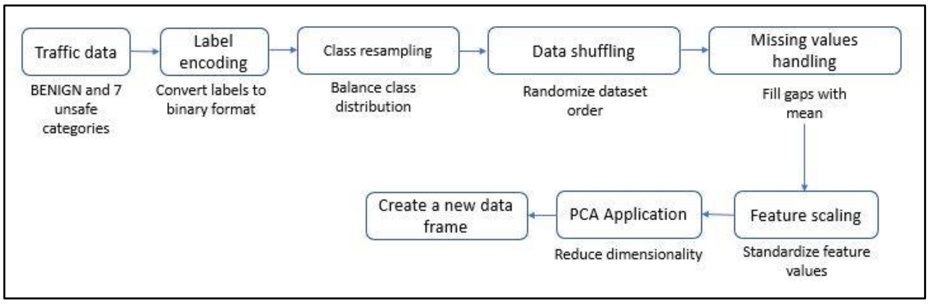

Figure 4: The stages of traffic data preprocessing

The last step after processing all the data is to downsize the data per category to get an equal number of samples. We accomplish this by taking the first min-size rows for each category, and then concat the balanced categories. As the page data contains the smallest number of examples (20,000), the size of all datasets was reduced to fit that limitation, resulting in 10,000 examples for each category. The indexes for URL_data, page_data, and traffic_data were also reset to avoid duplicating or overlapping records when merging.

All three datasets were merged horizontally, and features from each were integrated into a single horizontal table to prepare the data for analysis. Finally, the status of URL data, Email Type, and Label columns of traffic data were merged into a single column, final_label, which indicates what category each sample, belongs to. These stages are in Figure 5.

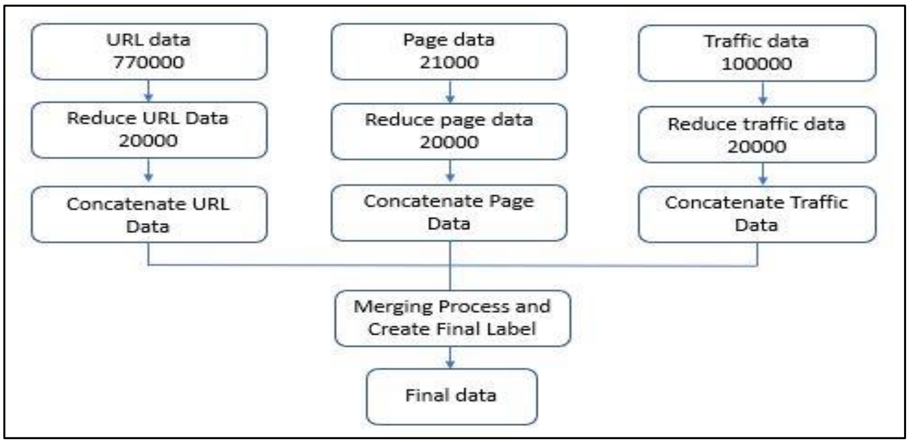

Figure 5: merged the three datasets

2- Adaptive Multi-Modal Phishing Detection Framework (AMPDF)

AMPDF (Adaptive Multi-Modal Phishing Detection Framework) is a browser plugin-based Phishing detection framework that features a three-dimensional data integration model to achieve a high degree of accuracy while maintaining a low error rate. The model consists of three core components, one for each kind of data related to a website. It happens in three phases:

Using CNNs for URL analyzing:

The model checks features related to URL, number of dots, number of dashes, URL length, and subdomain. Utilizing a Conv1D layer to extract critical features that could be considered suspicious of a phishing site, such as shortened URLs or suspicious domain structures. Then, it uses Global Max Pooling to reduce the dimensions and keep only the key features.
Deep Networks (Dense Layers) for Page Content Analysis
366 features corresponding to text, graphical elements, suspicious content, etc., are processed by the model. The Dense Layers help discover hidden patterns, using LeakyReLU rather than ReLU to avoid missing small yet significant information.
And for having Dense Layers: Traffic Behavior Analysis Using Deep Neural Networks
In this phase, 22 features related to user behavior during browsing are examined, including web traffic, clicks, and time-based behavior. For this purpose, the model uses a dense layer with ReLU activation to detect behavioral patterns that trigger phishing attempts.
test_size = 0.2, random_state = 42); # the test_data will be 20% of the entire data.
For url datatX of shape (8,) is reshaped to X of shape (8,1) for cnn. We used a Conv1D layer on 64 filters of size 3 and a GlobalMaxPooling1D layer after that in order to decrease dimensionality and keep useful features.
For the page data, which has 336 features, a Dense layer has been employed with LeakyRELU (alpha=0.1) activation to process features present in the web page to avoid vanishing gradients and grid search to speed up learning.
Then, a Dense layer with 32 neurons was used to extract behavioral patterns from the traffic data (22 features) and a Relu activation function to learn the non-linearities in traffic data. After going through these steps, Feature Fusion technology is used to combine the end results of the three subnetworks into a shared common layer. Then

the data is processed through the next neural layers (64 neurons) for further analysis. A Dropout layer 0.5 is added to prevent overfitting and improve generalization on unseen data. Finally, the processed data is then fed into a Sigmoid Output Layer to classify the website as fraudulent or legitimate. Adam and low learning rate (learning_rate=0.0001) were used for training to adjust model weights gradually. A data generator was used to keep only a portion of the data in memory at any time (a batch of 32). To ensure effective learning we train the model for 100 epochs. Figure 6 below presents the mechanism of AMPDF.

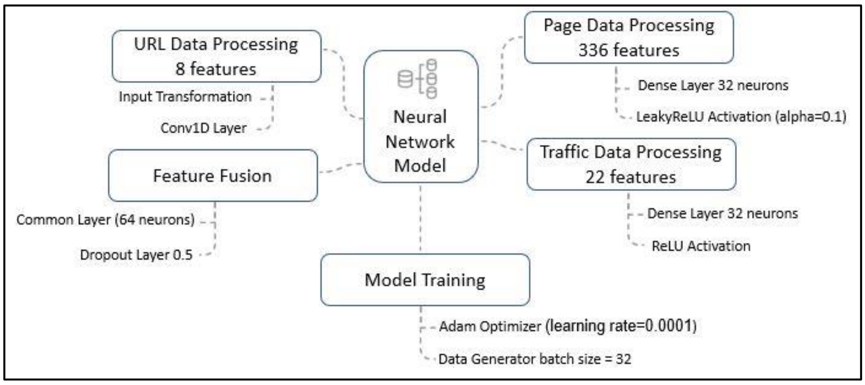

Figure 6: The mechanism of AMPDF

3-evaluation

A confusion matrix was then utilized as a means of assessing the performance of the proposed model, which included the distribution of correct and wrong predictions across the various categories. Metrics were moreover used to assess the performance of the model, such as accuracy, the percentage of correct predictions from total data; precision, the ability of the model to accurately predict positive classes; recall, the ability of the model to identify all actual positive classes; and F1 score, the harmonic mean of precision and recall, which is useful when the number of positive classes is much lower than negative classes. Furthermore, the loss for the model was calculated as the difference between the predictions and actual values.

Results and discussion

Metrics derived from the Confusion matrix, including Accuracy, Precision, Recall, and F1 score, were used to evaluate the model's performance on the data set. The outcome showed the high performance of the mode in classifying the data, reaching an accuracy of 98% in the test data. This indicates that the model seems to generalize well with no signs of overfitting. In order to understand the evolution of the model performance, a check was also made on its training progress. At first, the model started with an accuracy of 65.51% and a loss of 0.7383 at epoch1, which stated that the model was undertrained, which was the initial stage. By the second epoch, the accuracy had increased to 82.39%, indicating a fast learning process and a substantial improvement in performance. The fifth epoch achieved over 92% accuracy showing that the model was stable in learning. The last epoch (100) gave 99% accuracy, while the loss was 0.0360, meaning the model could discriminate between classes very effectively, as in Figure 7.

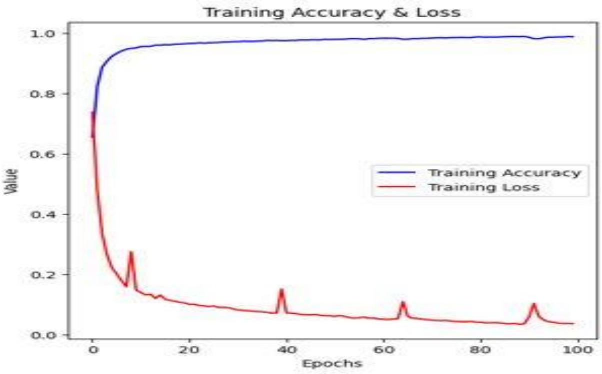

Figure 7: The accuracy and loss of training data

The model was then evaluated on both the training and test data, as shown below. The loss was 0.0708, which shows that the model is generalized well. It was 98% accurate, meaning no over-fitting. Accuracy refers to the proportion of true results among the total number of cases examined; therefore in this model, since the precision rate was 96.72%, it means that the model predicts very few cases as phishing that are not. The recall was 98.28%, showing that the model has a strong ability to detect actual phishing cases. The F1 score was 97.49%, confirming an excellent balance between precision and recall; the table 1 shows the accuracy metrics for the test data figure 8 shows the test loss and figure 9 displays the F1 score.

Table 1: The accuracy measures for the test data

accuracy	precision	Recall	F1 score
98%	96.7%	98.28	97.49

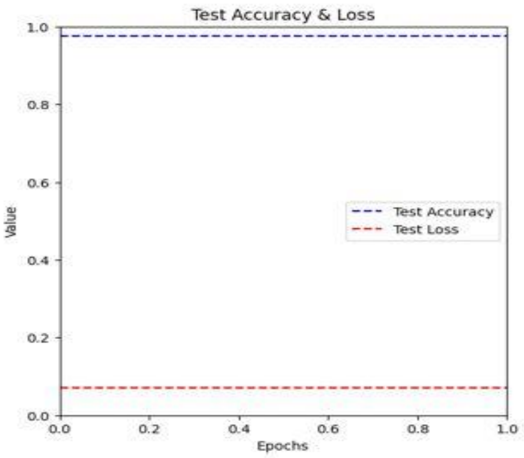

Figure 8: The accuracy and loss of testing data

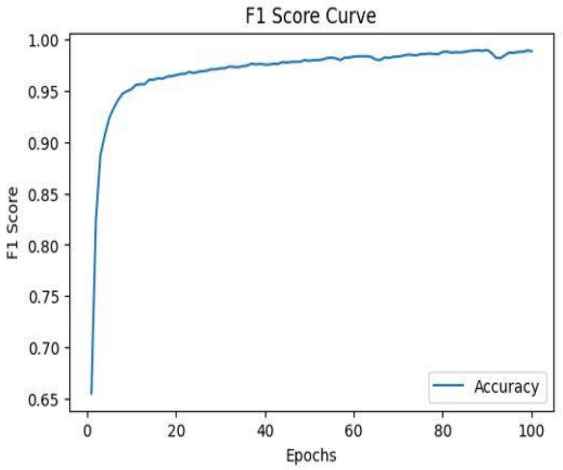

Figure 9: F1 score curve

Here is the confusion matrix to evaluate the model's prediction performance. It can be seen in the table above that the misclassifications are extremely few; only 66 cases from class 0 have been misclassified as class 1 and 34 instances from class 1 have been misclassified as class 0. Low False Positive Rate: Real cases should be detected as cases. In the same way, the False Negative ratio is low, which describes the model's high efficiency of detecting most of the phishing cases. The confusion matrix for test data is given in Table 2 and Figure 10.

Table 2: The confusion matrix

Predicted Class 1	Predicted Class 0	
66	1953	Actual class 0
1947	34	Actual class 1

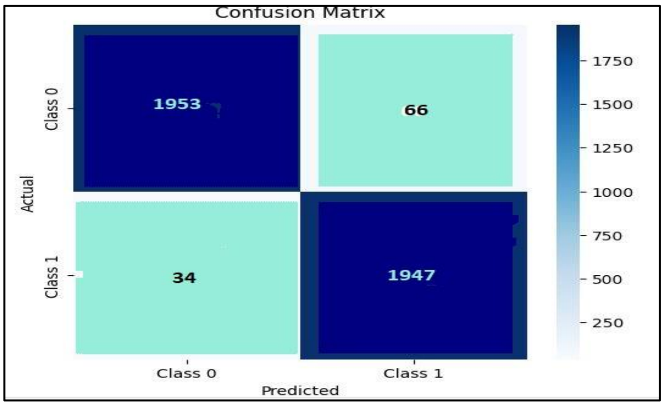

Figure 10: The confusion matrix

The comparative analysis performed is shown in Table 3 to analyze our proposed method as compared to previous work. We make this comparison with 14 recent studies, each using different techniques for phishing detection. On the other hand, though some of the models reached better accuracy at (>98%) margin, their inherent weakness lies in that they are concentrating on only one type of data, so they are too specific to work promptly on various kinds of phishing attacks.

Table 3: A comparative analysis between our proposed method with previous researchs

1	SI-BBA-A novel phishing website detection based on Swarm intelligence with deep learning	URL features	CNN	Accuracy: 94.8% 0.2 % False detections
2	An intelligent cyber security phishing detection system using deep learning techniques	Phishing Emails + Text Features	Boosted Decision Tree, SVM, NN	Acc 88.82% F1 score 88.74%
3	Phishing websites detection via CNN and multi-head self-attention on imbalanced datasets	URL features	CNN	Accuracy: 97.20% Recall= 95.60% Precision = 98.76% F1= 97.15%
4	Phishing URL Detection: A Real-Case Scenario Through Login URLs	PILU-90K dataset: 90K URLs, including homepage URLs, login URLs, and phishing sites	LR و TF-IDF و N-gram و CNN	Accuracy 96.50٪ F1 score 96.51٪
5	A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN	url	CNN, LSTM, LSTM-CNN	99.2%
6	Phishing detection system through hybrid machine learning based on URL	URL	Hybrid LSD model with LR, DT,SVM	Accuracy 98%
7	An enhanced deep learning-based phishing detection mechanism to effectively identify malicious URLs using variational autoencoders	URL	variational autoencoders (VAE) And deep neural network (DNN)	Accuracy 97.4%
8	DEPHIDES: Deep Learning Based Phishing Detection System	URLs	ANN, CNN, RNN, BRNN, ATT	Accuracy 98.74%
9	Advanced BERT and CNN-Based Computational Model for Phishing Detection in Enterprise Systems.	Emails	Bidirectional Encoder Representations from Transformers (BERT) for feature extraction and CNN for classification	accuracy 97.5%
10	OEC–Net: Optimal feature selection-based email classification network using unsupervised learning with deep CNN model	Email hybrid dataset incorporates UCI, CSDMC, and SpamAssassin information	principal component analysis (PCA)with Particle Swarm Optimization (PSO) to select features, and Deep Learning Convolutional Neural Network (DLCNN) model for classification	accuracy 98.43 %, precision 97.78 %, recall 96.41 %, and F1-score 97.07 %
11	Real-time phishing detection using deep learning methods by extensions	URLs	convolutional neural network (CNN), LR, DT, RF, SVM, and CNNLSTM	CNN Accuracy 98.4%
12	Enhancing Phishing Detection in Semantic Web Systems Using Optimized Deep Learning Models	emails	MobileBERT ،CMA-ES	Accuracy 95% F1 score 95%
13	Enhancing detection of zero-day phishing email attacks in the	emails	IndoBERT , FastText with CNN و LSTM	Accuracy 98.4375٪ F1 score 98%

	Indonesian language using deep learning algorithms			
14	Our proposed (AMPDF)	URL + Page Content + Traffic data sets	Hybrid (CNN + Dense Layers)	Test Loss: 0.07 Test Accuracy: 0.98% Test Precision: 0.97% Test Recall: 0.98% Test F1 Score: 0.97%

One of the studies obtained an accuracy of 99.2% by using a CNN+LSTM-CNN model, but this only depended on the URL links of the phishing. It is not possible to detect phishing based on other attack vectors such as content manipulation or behavioral deception. In gap, achieved 98.74% - 98.43% and OEC-Net accuracy, which were based on complex models such as BERT and PCA with PSO, have high resource usage, and both are unsuitable for some real-time use scenarios. Moreover, IndoBERT, which focused solely on email-based phishing detection, also achieved high accuracy (98.43%) but did not perform well on cross-site phishing and traffic-based deception techniques.

Instead, we proposed the AMPDF model, which employs a hybrid deep learning method that combines three data sources: URL features, content features and traffic features. This multi-source methodology improves the model's generalization across diverse phishing strategies rendering it more resilient against zero-day phishing attacks than models that utilize only URL data. Our approach looks not only at URL structure but also at content and behavioral patterns allowing it to adapt better to emerging phishing strategies.

Additionally, URL-based detection models are vulnerable to adversarial attacks in which phishing websites replicate existing structures to avoid being detected. This way, the vulnerability can be decreased and a more robust phishing detection framework can be constructed when combined both together by integrating content analysis with user behavior tracking methodologies [9].

One more restriction for a few high-accuracy models is their computational expenses. Methods like BERT-based algorithm feature extraction or PCA with PSO triple the processing time and require more system memory rendering them unviable for a real time detection system. Though our model has multiple data sources, we model it efficiently via an optimized CNN-based feature extraction mechanism and use dense layers to achieve maximum accuracy without compromising on speed.

CONCLUSION

We propose a hybrid framework, AMPDF that applies multi-source data analysis and deep learning to detect phishing attacks. This innovative approach which combines URL data, page content and traffic behavior, has led to an accuracy of 98% and a very low error rate demonstrating the model's ability to differentiate phishing sites from legitimate ones. The integration of convolutional neural networks (CNNs) and deep feature analysis provides superior phishing detection performance surpassing existing techniques based on blacklisting or manual feature engineering. Importantly, the proposed framework overcomes the practical challenges of training data generators high training costs and low performance, thereby offering a practical and effective solution for combating advanced phishing attacks. While our results are promising there are still challenges that need to be addressed. One such challenge is the need to generalize the model to previously unseen phishing attacks and enhance the efficiency of realtime detection. Therefore, we call for future studies to explore the potential of a hybrid approach that combines deep learning and natural language processing (NLP) for identifying advanced phishing techniques. We encourage researchers to focus their efforts on AI-based advanced phishing attack security as this is an area that holds great potential for further development and improvement.

REFRENCES

[1] S. Salloum, T. Gaber, S. Vadera, K. S.-P. C. Science, and undefined 2021, “Phishing email detection using natural language processing techniques: a literature survey,” ElsevierS Salloum, T Gaber, S Vadera, K ShaalanProcedia Computer Science, 2021•Elsevier, Accessed: Mar. 01, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921011741

[2] H. Alqahtani, S. Alotaibi, F. Alrayes, I. A.-T.-A. Sciences, and undefined 2022, “Evolutionary algorithm with deep auto encoder network based website phishing detection and classification,” mdpi.comH Alqahtani, SS

Alotaibi, FS Alrayes, I Al-Turaiki, KA Alissa, ASA Aziz, M MarayApplied Sciences, 2022•mdpi.com, Accessed: Mar. 01, 2025. [Online]. Available: https://www.mdpi.com/2076-3417/12/15/7441

[3] P. Bountakas, C. X.-J. of network and computer applications, and undefined 2023, "Helphed: Hybrid ensemble learning phishing email detection," ElsevierP Bountakas, C XenakisJournal of network and computer applications, 2023•Elsevier, Accessed: Mar. 01, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804522001862

[4] K. Thakur, M. Ali, M. Obaidat, A. K.- Electronics, and undefined 2023, "A systematic review on deep-learning-based phishing email detection," mdpi.comK Thakur, ML Ali, MA Obaidat, A KamruzzamanElectronics, 2023•mdpi.com, Accessed: Mar. 01, 2025. [Online]. Available: https://www.mdpi.com/2079-9292/12/21/4545

[5] H. F. Atlam and O. Oluwatimilehin, "Business Email Compromise Phishing Detection Based on Machine Learning: A Systematic Literature Review," Jan. 01, 2023, MDPI. doi: 10.3390/electronics12010042.

[6] A. Al Tawil, L. Almazaydeh, ... D. Q.-Comput. M., and undefined 2024, "Comparative Analysis of Machine Learning Algorithms for Email Phishing Detection Using TF-IDF, Word2Vec, and BERT," researchgate.netA Al Tawil, L Almazaydeh, D Qawasmeh, B Qawasmeh, M Alshinwan, K ElleithyComput. Mater. Contin, 2024•researchgate.net, Accessed: Mar. 01, 2025. [Online]. Available: https://www.researchgate.net/profile/Arar-Tawil-2/publication/385775640_Comparative_Analysis_of_Machine_Learning_Algorithms_for_Email_Phishing_Detection_Using_TF-IDF_Word2Vec_and_BERT/links/673b7a2bc1b80e561646066a/Comparative-Analysis-of-Machine-Learning-Algorithms-for-Email-Phishing-Detection-Using-TF-IDF-Word2Vec-and-BERT.pdf

[7] D. Kalla, F. Samaah, S. Kuraku, N. S.-I. J. of, and undefined 2023, "Phishing detection implementation using databricks and artificial Intelligence," academia.eduD Kalla, F Samaah, S Kuraku, N SmithInternational Journal of Computer Applications, 2023•academia.edu, vol. 185, no. 11, pp. 975–8887, 2023, Accessed: Mar. 01, 2025. [Online]. Available: https://www.academia.edu/download/110809011/ijca2023922764.pdf

[8] P. Bountakas, K. Koutroumpouchos, and C. Xenakis, "A Comparison of Natural Language Processing and Machine Learning Methods for Phishing Email Detection," ACM International Conference Proceeding Series, Aug. 2021, doi: 10.1145/3465481.3469205.

[9] M. Al Fayoumi, A. Odeh, ... I. K.-2022 I. 12th, and undefined 2022, "Email phishing detection based on naïve Bayes, Random Forests, and SVM classifications: A comparative study," ieeexplore.ieee.orgM Al Fayoumi, A Odeh, I Keshta, A Aboshgifa, T AlHajahjeh, R Abdulraheem2022 IEEE 12th Annual Computing and Communication Workshop and, 2022•ieeexplore.ieee.org, Accessed: Mar. 05, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9720757/

[10] S. Arya and S. Chamotra, "Multi Layer Detection Framework for Spear-Phishing Attacks," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 13146 LNCS, pp. 38–56, 2021, doi: 10.1007/978-3-030-92571-0_3.

[11] H. Fares, N. Mouakkal, ... Y. B.-I. J. of, and undefined 2024, "Robust email phishing detection using machine learning and deep learning approach," search.proquest.comH Fares, N Mouakkal, Y Baddi, N HajraouiInternational Journal of Communication Networks and Information, 2024•search.proquest.com, Accessed: Mar. 05, 2025. [Online]. Available: https://search.proquest.com/openview/363fd4818d3f20d5390fdfb864a316f5/1?pq-origsite=gscholar&cbl=52057

[12] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, "Deep Learning for Phishing Detection: Taxonomy, Current Challenges and Future Directions," 2022, Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/ACCESS.2022.3151903.

[13] R. Chataut, P. Gyawali, Y. U.-2024 I. 14th Annual, and undefined 2024, "Can ai keep you safe? a study of large language models for phishing detection," ieeexplore.ieee.orgR Chataut, PK Gyawali, Y Usman2024 IEEE 14th Annual Computing and Communication Workshop and, 2024•ieeexplore.ieee.org, Accessed: Mar. 05, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10427626/

[14] A. Muir, K. Brown, and A. Girma, "Reviewing the Effectiveness of Multi-factor Authentication (MFA) Methods in Preventing Phishing Attacks," Lecture Notes in Networks and Systems, vol. 1157 LNNS, pp. 597–607, 2024, doi: 10.1007/978-3-031-73128-0_40.

[15] A. Abbas, "Evolving Phishing Defense: Innovative Defense Mechanisms and Effective Measurement Strategies," 2024, Accessed: Mar. 05, 2025. [Online]. Available: https://www.researchgate.net/profile/Asad-Abbas-41/publication/384355762_Evolving_Phishing_Defense_Innovative_Defense_Mechanisms_and_Effective_

Measurement_Strategies/links/66f564f0869f1104c6b5b592/Evolving-Phishing-Defense-Innovative-Defense-Mechanisms-and-Effective-Measurement-Strategies.pdf

[16] & C. S.-G. I. J. of E. and undefined 2024, "A Comprehensive Analysis on Multi-Layered Machine Learning Approaches for Detecting and Preventing Phishing in Email and Websites.," search.ebscohost.comCE ShyniGrenze International Journal of Engineering & Technology (GIJET), 2024•search.ebscohost.com, Accessed: Mar. 05, 2025. [Online]. Available: https://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=23955287&AN=181715185&h=E%2F1RPj%2B36jjc7RgHYdvWSNJw004Q6UYgrQ1PF5ZqUcl7Ny%2BYB700j1DsMLjhHI19p70hCdxMruRzoCImC2bm0Q%3D%3D&crl=c

[17] O. Sahingoz, S. Baykal, D. B.-& I. T. (CS & IT), and undefined 2018, "Phishing detection from urls by using neural networks," csitcp.comOK Sahingoz, SI Baykal, D BulutComputer Science & Information Technology (CS & IT), 2018•csitcp.com, Accessed: Mar. 05, 2025. [Online]. Available: https://csitcp.com/paper/8/817csit05.pdf

[18] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. S.-C. Networks, and undefined 2020, "Accurate and fast URL phishing detector: a convolutional neural network approach," ElsevierW Wei, Q Ke, J Nowak, M Korytkowski, R Scherer, M WoźniakComputer Networks, 2020•Elsevier, Accessed: Mar. 05, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128620301109

[19] A. Md, D. Jaiswal, J. Daftari, S. Haneef, C. I.- Electronics, and undefined 2022, "Efficient dynamic phishing safeguard system using neural boost phishing protection," mdpi.comAQ Md, D Jaiswal, J Daftari, S Haneef, C Iwendi, SK JainElectronics, 2022•mdpi.com, Accessed: Mar. 05, 2025. [Online]. Available: https://www.mdpi.com/2079-9292/11/19/3133

[20] M. Hussain, C. Cheng, R. Xu, and M. Afzal, "CNN-Fusion: An effective and lightweight phishing detection method based on multi-variant ConvNet," Inf Sci (N Y), vol. 631, pp. 328–345, Jun. 2023, doi: 10.1016/J.INS.2023.02.039.

[21] L. Halgaš, I. Agrafiotis, and J. R. C. Nurse, "Catching the Phish: Detecting Phishing Attacks Using Recurrent Neural Networks (RNNs)," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11897 LNCS, pp. 219–233, 2020, doi: 10.1007/978-3-030-39303-8_17.

[22] U. Joseph, M. J. I. J. of Engineering, and undefined 2021, "Real Time Detection of Phishing Attacks in Edge Devices," academia.edu, Accessed: Mar. 05, 2025. [Online]. Available: https://www.academia.edu/download/68314576/real_time_detection_of_phishing_IJERTCONV9IS07025.pdf

[23] M. Sameen, K. Han, S. H.-I. Access, and undefined 2020, "PhishHaven—An efficient real-time AI phishing URLs detection system," ieeexplore.ieee.orgM Sameen, K Han, SO HwangIeee Access, 2020•ieeexplore.ieee.org, Accessed: Mar. 05, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9082616/

[24] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, J. N.- Electronics, and undefined 2020, "An effective phishing detection model based on character level convolutional neural network from URL," mdpi.comA Aljofey, Q Jiang, Q Qu, M Huang, JP NiyigenaElectronics, 2020•mdpi.com, Accessed: Mar. 05, 2025. [Online]. Available: https://www.mdpi.com/2079-9292/9/9/1514

[25] A. Ozcan, C. Catal, E. Donmez, B. S.-N. C. and, and undefined 2023, "A hybrid DNN–LSTM model for detecting phishing URLs," SpringerA Ozcan, C Catal, E Donmez, B SenturkNeural Computing and Applications, 2023•Springer, vol. 35, no. 7, pp. 4957–4973, Mar. 2023, doi: 10.1007/s00521-021-06401-z.