JOGHRM

International Journal of Information Engineering and Science

e-ISSN : 3048-1953
p-ISSN : 3048-1902

Research Article

Detecting Phishing URLs with CNN - Decision Tree Method

Reza Aminullah[1]*, Fetty Tri Anggraeny[2], Fawwaz Ali Akbar[3]

[1-3] University Veteran Jawa Timur ; e-mail : aminullahreza65@gmail.com
* Corresponding Author : Reza Aminullah

Abstract: This research focuses on assessing the efficacy of a method that integrates Convolutional Neural Networks (CNN) with Decision Trees for the detection of phishing URLs. Phishing represents a major cyber threat, where cybercriminals attempt to deceive individuals into disclosing sensitive information via fraudulent websites. As the frequency of phishing attacks continues to rise, there is a pressing need for effective detection and prevention strategies. In this investigation, a dataset comprising both phishing and legitimate URLs was utilized to train a CNN-Decision Tree model. The training phase includes feature extraction from URLs using CNN, which excels at identifying intricate patterns within the data, followed by classification through Decision Trees, recognized for their capacity to deliver straightforward and comprehensible interpretations of classification outcomes. The model's performance was evaluated across nine distinct scenarios to assess its effectiveness under varying conditions. The results indicated that the hybrid CNN-Decision Tree model achieved a precision rate of 94%, a recall of 90%, and an F1-Score of 92%, with an overall accuracy of 93%. These findings suggest that the model is not only proficient in identifying phishing URLs but also maintains a commendable balance between precision and recall. This research highlights that the synergy of CNN and Decision Trees can serve as a potent solution for phishing URL detection, significantly contributing to the advancement of enhanced cybersecurity systems.

Keywords: Phishing, Detection, URL, Network, Learning.

1. Introduction

In the current fast-evolving digital landscape, cybersecurity threats have emerged as a significant global issue, including in Indonesia. Among these threats, phishing stands out as particularly alarming, as attackers seek to obtain sensitive information through fraudulent websites. With the advancement of technology and the growing prevalence of internet usage, phishing attacks are becoming increasingly sophisticated and harder to identify. These attacks target not only individuals but also large organizations, leading to substantial financial repercussions.

The fourth industrial revolution has ushered in transformative changes across various facets of life, particularly in data management and protection. As devices become more interconnected through the Internet of Things (IoT), the amount of data generated has surged dramatically. This surge presents new challenges for data security, as threats like phishing can take advantage of vulnerabilities within interconnected systems.

In this scenario, machine learning technologies present a promising avenue for solutions. Methods such as Convolutional Neural Networks (CNN) and Decision Trees have demonstrated their effectiveness in numerous classification tasks. CNNs excel at extracting intricate features from data, while Decision Trees are valued for their capacity to offer clear and comprehensible interpretations of classification outcomes (Quinlan, 1986; Sultana et al.,

Received: February 12th, 2025
Revised: February 28th, 2025
Accepted: March 05th, 2025
Online Available : March 07th, 2025
Curr. Ver.: March 07th, 2025

2019). The integration of these two methodologies into a hybrid CNN-Decision Tree model is anticipated to enhance the accuracy of phishing URL detection.

This research aims to assess the efficacy of the hybrid CNN-Decision Tree model in identifying phishing URLs. Utilizing a dataset that includes both phishing and legitimate URLs, the study will evaluate the model's effectiveness across various scenarios. The testing process is designed to explore how this model can be applied in real-world phishing detection systems and to identify potential challenges that may arise.

Data security remains a critical concern in this digital age. Given the escalating threat of phishing, comprehensive research is essential to develop more effective detection methods. This study concentrates on employing a hybrid CNN-Decision Tree model to bolster data security and safeguard users against increasingly intricate phishing attacks. By gaining insights into the workings of this model, it is hoped that organizations and individuals can better protect themselves from sophisticated phishing threats.

Furthermore, this research aims to offer new perspectives in the realm of cybersecurity, particularly in creating a more efficient and reliable phishing detection system. The findings are expected to serve as a reference for future studies and contribute to the overall enhancement of cybersecurity. The simultaneous application of CNN and Decision Tree methods within a hybrid model holds significant promise for detecting phishing URLs (Bagui et al., 2021).

2. Research Methods

Research Design

This research focuses on a combined approach utilizing a Convolutional Neural Network (CNN) and a Support Vector Machine (SVM) for the detection of phishing URLs. The objective of this study is to evaluate and compare the effectiveness of these two methodologies in recognizing URLs that may pose a threat. By employing a curated dataset, the study will conduct tests and analyze the outcomes of the hybrid model's detection capabilities.Dataset

The dataset utilized in this research comprises two distinct categories of URLs: phishing and non-phishing. This dataset is sourced from a reputable platform, kaggle.com, which offers data for various research purposes. The process of data collection involves selecting verified URLs to ensure that the information used in this study is both accurate and pertinent. The dataset is split into two segments: training data and testing data, with multiple scenarios for data division, including 90% for testing and 10% for training, 80% for testing and 20% for training, and 70% for testing and 30% for training. The training data is employed to develop the model, while the testing data is used to assess the model's performance following the training phase.

Data Preprocessing

Prior to utilizing the data for training, a crucial preprocessing phase is conducted to prepare the dataset. This step is essential to ensure that the model can effectively learn from the provided information. The preprocessing involves several key stages, which include:

- Lemmatization: This stage involves converting words to their root forms. While not all components of URLs require lemmatization, certain frequently occurring keywords do. For instance, the term "phishing" can be simplified to its base form, and variations like "login" and "logins" can be standardized to "login." This simplification reduces data complexity and enhances the model's ability to identify relevant patterns.

- Tokenization: Following lemmatization, the next phase is tokenization, which entails breaking down a URL into smaller segments or tokens for further analysis. In the context of URLs, tokens may consist of parts separated by specific characters, such as slashes (/) or question marks (?). Tokenization aids in pinpointing significant elements within a URL that are pertinent to phishing detection.

- Padding and Reshaping: After tokenization, the data undergoes padding and reshaping. Padding ensures that all inputs to the model are uniform in length, which may involve adding blank characters to shorter URLs to match the length of the longest URL in the dataset. Reshaping is also necessary to adjust the data dimensions to align with the input requirements of the CNN model. This step guarantees that the data can be processed accurately by the model without errors.

- Normalization: The final preprocessing step is normalization, which standardizes all features to a common scale. For example, the length of a URL can be normalized to a range of [0, 1] to prevent any single feature from dominating the training process. This normalization is vital for enhancing model performance and accelerating convergence during training. By normalizing the data, the model can learn more efficiently and yield more precise predictions.

By completing these preprocessing steps, the training data becomes cleaner, more organized, and more informative, enabling the model to learn more effectively and improving its accuracy in detecting phishing URLs.

Hybrid CNN - Decission Tree model

- Decision Tree hybrid model is built with two main stages:

1. Training the CNN: A Convolutional Neural Network (CNN) model is developed using the training dataset to identify features within URLs. The training phase starts with the specification of the maximum input length, which is determined by the padding size applied earlier. The architecture of the CNN comprises multiple layers, including an embedding layer, a convolutional layer, a pooling layer, and a fully connected layer..

   - Embedding Layer : This layer is responsible for transforming word representations into low-dimensional vectors. Through the use of embedding, the model is able to grasp the semantic significance of words within the URL, which is crucial for subsequent analysis.

   - Convolution Layer: The convolutional layer plays a crucial role in identifying local patterns within the data. It employs filters that slide across the input to recognize significant features, such as sequences of characters or combinations of words that are commonly found in phishing URLs. The convolution process utilizes what is called a filter. Like an image, a filter has a certain height, width, and thickness. This filter is initialized with a certain value, and the value of this filter is the parameter that will be updated in the learning process." (Pangestu, R. A., Fetty, T. A, & Rahmat, B.., 2020). Additionally, the implementation of the ReLU (Rectified Linear Unit) activation function in this layer enhances the training speed and boosts the overall performance of the model.

   - Pooling Layer : Following the convolution layer, a pooling layer is utilized to decrease the data's dimensionality. This step not only minimizes the number of parameters that need to be learned but also mitigates the risk of overfitting by eliminating extraneous information. As a result, the model can concentrate more effectively on the most significant features..

   - Dropout Layer : To further mitigate the risk of overfitting, a dropout layer is incorporated following the pooling layer and preceding the fully connected layer. This layer randomly deactivates a portion of the neurons during the training process, compelling the model to develop more resilient and generalizable representations..

- Fully Connected Layer : Once the features have been obtained through the convolution and pooling layers, the fully connected layer integrates all the neurons to generate the final output. This layer plays a crucial role in classifying URLs according to the features that have been extracted. By employing the sigmoid activation function in the output layer, the model is able to deliver probabilities for two categories: phishing and non-phishing..

- The CNN model is trained over several epochs, The training data is split into training and validation sets. This approach enables the model to learn from the data, enhancing its ability to accurately identify phishing URLs. After the training phase is finished, features from both the training and test datasets are extracted to be utilized in the subsequent classification stage.

2. Classification with Decision Tree: After the features are extracted, the results from the CNN are used as input for the Decision Tree model. The Decision Tree will perform classification based on the extracted features, separating phishing URLs from non-phishing URLs. This process involves forming a decision tree structure that describes a set of rules based on relevant features. Each node in the tree represents a feature used to divide the data, while the branches show the results of the decisions.

   The decision to divide the data at each node is made based on certain criteria, such as Gini impurity or entropy, which aims to maximize the information obtained from each division. This process continues until it reaches the leaf node, which represents the final class (phishing or non-phishing). This Decision Tree model provides a clear and easy-to-understand interpretation of the classification decision, and can capture patterns in the data to provide relevant classification results based on the features that have been extracted.

Model Testing and Evaluation

After the model is trained, testing is performed using the test data to evaluate the model's performance. Some of the evaluation metrics used in this study include:

- Accuracy: This metric represents the proportion of accurate predictions relative to the total number of predictions made. It offers a general assessment of the model's effectiveness in classifying data.

- Precision: This is the ratio of true positive predictions to the total number of positive predictions. Precision is crucial for evaluating how many of the predicted positive cases are indeed phishing URLs.

- Recall : This metric indicates the ratio of true positive predictions to the total number of actual positive instances. Recall assesses the model's capability to identify all phishing URLs effectively.

- F1-Score: The harmonic mean of precision and recall, giving a better idea of the balance between the two. F1-Score is particularly useful when there is an imbalance between the positive and negative classes.

Development Tools and Environment

This study was carried out using the Python programming language, utilizing machine learning libraries like TensorFlow and Scikit-learn. The development environment chosen for this research was Jupyter Notebook, which enables interactive testing and visualization of outcomes. Furthermore, this tool streamlines the process of model development and testing, making it easier for researchers to conduct data analysis.

Research Procedures

The research procedure is carried out in several systematic steps to ensure that each stage is carried out properly and the results obtained are reliable. These steps are as follows:

1. Data Collection: The initial phase of the research process involves gathering a pertinent dataset, which includes both phishing and non-phishing URLs sourced from reliable origins. Once the data is collected, the subsequent step is to verify its accuracy, ensuring that all URLs in the dataset are confirmed and appropriately categorized. This validation is crucial to prevent any inaccuracies during model training that could impact the final outcomes.

2. Preprocessing: Once the data has been gathered, the following step is to perform preprocessing. This phase encompasses various stages, including lemmatization, tokenization, padding and reshaping, normalization, and feature extraction. Each of these stages is designed to make the data cleaner, more organized, and more informative, enabling the model to learn more efficiently. Effective preprocessing enhances data quality, which subsequently boosts the performance of the model.

3. Model Training: Once preprocessing is finished, the subsequent step is to train the model. The CNN model is trained on the training data to identify features from the URLs. This training phase includes configuring model parameters and applying optimization algorithms to enhance accuracy. After the CNN model has been trained, the results of the feature extraction are utilized as input for the SVM model, which will carry out classification based on these extracted features. The goal of this process is to develop a model capable of effectively distinguishing between different categories.

4. Model Testing: After the model has been trained, the next phase is to test it. The test data that was prepared earlier is utilized to assess the model's performance. This evaluation involves employing metrics such as accuracy, precision, recall, and F1-Score to determine how effectively the model can identify phishing URLs. The results from this testing will offer insights into the effectiveness of the developed model.

Research result

Model Performance

Following the training and testing phases, the hybrid Convolutional Neural Network (CNN) and Decision Tree model was assessed using various performance metrics. The results indicate that this model performs well in identifying phishing URLs. The table below provides a summary of the evaluation outcomes from nine distinct scenarios :

Table 1

Scenario	Precision	Recall	F1-Score	Accuracy
1	0.86	0.86	0.86	86%
2	0.89	0.90	0.89	88%
3	0.83	0.85	0.83	86%
4	0.85	0.89	0.86	86%
5	0.86	0.94	0.87	87%
6	0.86	0.89	0.87	87%
7	0.96	0.94	0.95	95%
8	0.86	0.89	0.87	87%
9	0.96	0.94	0.94	95%
Average				88%

From the table above, it can be seen that the model shows variation in performance based on the scenarios tested.

- Precision: Precision values range from 0.83 to 0.96. Scenarios 7 and 9 show the highest precision of 0.96, indicating that out of all the positive predictions made by the model, 96% of them are actually phishing URLs. This shows that the model is very effective in reducing false positives.

- Recall: The recall value also shows good performance, with the highest value of 0.94 in scenarios 5, 7, and 9. High recall indicates that the model is able to detect most of the phishing URLs in the dataset, with few missed.

- F1-Score: The F1-Score, which is the harmonic mean of precision and recall, ranges from 0.83 to 0.95. Scenario 7 has the highest F1-Score of 0.95, indicating a good balance between precision and recall. This is important in the context of phishing detection, where both false positives and false negatives must be minimized.

- Accuracy: The overall accuracy of the model ranges from 86% to 95%. Scenarios 7 and 9 show the highest accuracy of 95%, indicating that the model was able to correctly classify 95 out of 100 tested URLs.

Error Analysis

Although the model performs well, there are some errors that need to be analyzed. Some phishing URLs that are not detected by the model often have a structure similar to legitimate URLs, making it difficult for the model to distinguish them. For example, URLs that use domains that are very similar to legitimate domains or URLs that have a reasonable length and do not contain suspicious keywords.

Practical Implications

The results of this study have significant practical implications in the field of cybersecurity. By using a hybrid CNN-Decision Tree model, this merger can improve their ability to detect and prevent phishing attacks. The implementation of this model-based detection system can help protect sensitive data and prevent financial losses caused by phishing attacks.

3. Discussion

Model Performance Analysis

The results show that the hybrid Convolutional Neural Network (CNN) and Decision Tree model has good performance in detecting phishing URLs. Based on the results table, this model achieves consistent precision, recall, and F1-Score across scenarios. The average model precision is in the range of 0.86 to 0.96, indicating that the model is able to identify phishing URLs with a low error rate.

The Influence of Scenarios on Performance

Of the nine scenarios tested, it can be seen that the scenarios with the highest precision (0.96) and accuracy (95%) occur in scenarios 7 and 9. This shows that variations in training and testing data, as well as the parameters used in the model, can significantly affect performance. Scenarios with a larger proportion of training data tend to produce better results, indicating the importance of sufficient data for model training.

Balance Between Precision and Recall

The model shows a good balance between precision and recall, with F1-Score values varying between 0.83 and 0.95. This balance is important in the context of phishing detection, where both false positives and false negatives can have serious consequences. For example, a false positive can cause a user to lose access to a legitimate service, while a false negative can result in the theft of sensitive data.

Error Analysis

Although the model performed well, there were some errors that needed to be analyzed. Some undetected phishing URLs often had structures similar to legitimate URLs, making it difficult for the model to distinguish them. For example, URLs that use domains that are very similar to legitimate domains or URLs that have a reasonable length and do not contain suspicious keywords. Analysis of these errors provides important insights for future model development.

Implications for Further Research

The results of this study provide a strong foundation for further research in phishing detection. Future studies can explore the use of other deep learning techniques, such as Recurrent Neural Networks (RNN) or Long Short-Term Memory (LSTM), which may be more effective in handling sequential data such as URLs. In addition, studies can consider using larger and more diverse datasets to improve model generalization.

4. Conclusion

This study has successfully developed and evaluated a hybrid model of Convolutional Neural Network (CNN) and Decision Tree to detect phishing URLs. Based on the results obtained, several conclusions can be drawn as follows:

1. Model Performance: The hybrid CNN-Decision Tree model shows good performance with consistent precision, recall, and F1-Score across scenarios. Test results show that the model is able to detect phishing URLs with high accuracy, reaching up to 95% in some scenarios.

2. Effectiveness of the Method: The use of CNN for feature extraction from URLs proved effective in capturing complex patterns present in the data. Decision Tree then successfully classified URLs based on the extracted features, indicating that the combination of these two methods can improve detection accuracy.

3. Performance Stability: The model shows good performance stability across scenarios, with similar precision and recall values. This indicates that the model is reliable in detecting phishing URLs, despite variations in the data used.

4. Practical Implications: The results of this study have significant practical implications in the field of cybersecurity. By using this hybrid model, organizations can improve their ability to detect and prevent phishing attacks, protect sensitive data, and reduce financial losses caused by such attacks.

5. Recommendations for Further Research: This study provides a solid foundation for further research in phishing detection. Further research can explore the use of other deep learning techniques and use larger datasets to improve model generalization.

Thus, this study shows that the hybrid CNN-Decision Tree model is an effective tool in detecting phishing URLs, and the results can contribute to the development of better cybersecurity systems.

References

[1]. Al-Sartawi, A. M. A. M. (2020). Information technology governance and cybersecurity at the board level. International Journal of Critical Infrastructures, 16(2), 150–161. https://doi.org/10.1504/ijcis.2020.10029173

[2]. APWG. (n.d.). Phishing e-mail reports and phishing site trends. Retrieved from https://www.apwg.org

[3]. Barik, K., Misra, S., & Mohan, R. (2025). Web-based phishing URL detection model using deep learning optimization techniques. International Journal of Data Science and Analytics. https://doi.org/10.1007/s41060-025-00728-9

[4]. Barik, K., Misra, S., & Sanz, L. F. (2024). A model for estimating resiliency of AI-based classifiers defending against cyber attacks. International Journal of Computational Intelligence Systems, 17(1), 1–15. https://doi.org/10.1007/s44196-024-00686-3

[5]. Fazeldehkordi, E. (2014). A machine learning approach to phishing detection and defense. Retrieved from https://www.researchgate.net/publication/267156776

[6]. Goel, D., & Jain, A. K. (2018). Mobile phishing attacks and defence mechanisms: State of art and open research challenges. Computer Security, 74, 120–133. https://doi.org/10.1016/j.cose.2017.12.006

[7]. Greene. (2018). No phishing beyond this point. IEEE Computing, 58(7), 67–75. https://doi.org/10.1109/MC.2018.2701632

[8]. Huang, K., Madnick, S. E., & Johnson, S. (2020). Framework for understanding cybersecurity impacts on international trade. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3555341

[9]. Ikeda, K., Marshall, A., & Zaharchuk, D. (2019). Agility, skills and cybersecurity: Critical drivers of competitiveness in times of economic uncertainty. Strategic Leadership, 47(3), 40–48. https://doi.org/10.1108/SL-02-2019-0032

[10]. Kavya, S., & Sumathi, D. (2025). Staying ahead of phishers: A review of recent advances and emerging methodologies in phishing detection. Artificial Intelligence Review, 58(2), 329–350. https://doi.org/10.1007/s10462-024-11055-z

[11]. Maware, C., Parsley, D. M., Huang, K., Swan, G. M., & Akafuah, N. (2023). Moving lab-based in-person training to online delivery: The case of a continuing engineering education program. Journal of Computer Assisted Learning, 39(4), 1167–1183. https://doi.org/10.1111/jcal.12789

[12]. Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1), 81–106. https://doi.org/10.1007/BF00116251

[13]. Sahingoz, O. K., Buber, E., Demir, O., & 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

[14]. Yang, P., Zhao, G., & Zeng, P. (2019). Phishing website detection based on multidimensional features driven by deep learning. IEEE Access, 7, 15196–15209. https://doi.org/10.1109/ACCESS.2019.2892066

[15]. Lallie, H. S., et al. (2021). Cybersecurity in the age of COVID-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic. Computers & Security, 105, 102248. https://doi.org/10.1016/j.cose.2021.102248