This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3348071

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022.Doi Number

Multi-modal Features Representation-based Convolutional Neural Network Model for Malicious Website Detection

Mohammed Alsaedi [1], Fuad A. Ghaleb [2,*], Faisal Saeed [3,*], Member, IEEE, Jawad Ahmad [4] and Mohammed Alasli [2]

[1]College of Computer Science and Engineering, Taibah University, P.O.Box. 344, Medina 41411, Western Region, Saudi Arabia;
[2]Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru 81310, Johor, Malaysia
[3]DAAI Research Group, Department of Computing and Data Science, School of Computing and Digital Tech-nology, Birmingham City University, Birmingham B4 7XG, UK
[4]School of Computing, Engineering and the Built Environment, Edinburgh Napier University, UK

Corresponding author: Fuad A. Ghaleb (e-mail: abdulgaleel@utm.my) and Faisal Saeed (e-mail: faisal.saeed@bcu.ac.uk).

This research was funded by the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia under the project number (168/442)

ABSTRACT Web applications have proliferated across various business sectors, serving as essential tools for billions of users in their daily lives activities. However, many of these applications are malicious which is a major threat to Internet users as they can steal sensitive information, install malware, and propagate spam. Detecting malicious websites by analyzing web content is ineffective due to the complexity of extraction of the representative features, the huge data volume, the evolving nature of the malicious patterns, the stealthy nature of the attacks, and the limitations of traditional classifiers. Uniform Resource Locators (URL) features are static and can often provide immediate insights about the website without the need to load its content. However, existing solutions for detecting malicious web applications through web content analysis often struggle due to complex feature extraction, massive data volumes, evolving attack patterns, and limitations of traditional classifiers. Leveraging solely lexical URL features proves insufficient, potentially leading to inaccurate classifications. This study proposes a multimodal representation approach that fuses textual and image-based features to enhance the performance of the malicious website detection. Textual features facilitate the deep learning model's ability to understand and represent detailed semantic information related to attack patterns, while image features are effective in recognizing more general malicious patterns. In doing so, patterns that are hidden in textual format may be recognizable in image format. Two Convolutional Neural Network (CNN) models were constructed to extract the hidden features from both textual and image-represented features. The output layers of both models were combined and used as input for an artificial neural network classifier for decision-making. Results show the effectiveness of the proposed model when compared to other models. The overall performance in terms of Matthews Correlation Coefficient (MCC) was improved by 4.3% while the false positive rate was reduced by 1.5%.

INDEX TERMS Convolutional Neural Network, Malicious URL detection, Malicious Website Detection, Multi-modal Features Representation, URL image representation.

I. INTRODUCTION

According to the Siteefy website[1] , there are over 1.11 billion websites in the World, and this number has been growing exponentially in recent years. Every day, T 252 thousand new websites are created (REF Please). As of May 9, 2023, it is estimated that the number of web pages is more than 50 billion pages. Although most of the websites are created for good purposes, many of these websites are malicious websites [2]. Malicious websites are designed to harm users in some way, such as by stealing their personal information or installing malware on their computers. They can be used to spread malware, phishing, spread spam, or conduct denial of service attacks [3]. According to Google's in-depth research, there are an estimated 12.8 million malicious websites on the internet [4]. Furthermore, as stated by authors in [5], there are 18.5

million websites hosting malicious code. This number is constantly changing, as new malicious websites are created and old ones are taken down.

Malicious website detection has been the subject of much research and many solutions were suggested [6-23]. The blacklist is the most common solution used by many organizations [24]. However, it is slow to update, as malicious actors can easily bypass blacklists by creating new websites or simply changing the URLs of their websites. This makes it difficult for blacklist-based systems to keep up with the ever-changing landscape of malicious websites [25, 26].

To address the limitations of blacklisting, many researchers have employed machine learning techniques to detect malicious websites. These techniques extract features from web content [27-29], scripts [15, 16], HTTP/s response [29, 30], URLs [6-14, 31-33], domain names [25, 34, 35], network traffic data [34, 36], and digital certificates [26]. Many machine learning algorithms were used such as support vector machines, decision trees, logistic regression, and random forests to classify websites as malicious or benign [28, 32]. The effectiveness of machine learning methods depends on the choice of features [13, 14, 17-23]. However, extracting effective features is challenging due to the constant changing of malicious code, the use of obfuscation techniques by attackers, the huge volume of data that needs to be analyzed, and the complexity of the attack today. Unfortunately, traditional machine learning is ineffective in extracting useful patterns for classification from huge and complex datasets. However, effective feature engineering is required to improve detection performance.

Deep learning models are effective in extracting representative features from huge and complex datasets. They can automatically extract effective features without the need for incentive manual feature engineering, as it can automatically learn features from webpage text data. Convolutional Neural Networks (CNN) [22], Recurrent Neural Networks (RNN) [23], and attention mechanisms were commonly reported methods for malicious malware detection. Many deep learning models are constructed based on features extracted from the website's content. However, acquiring large and diverse datasets from website content for training deep learning models is challenging due to the dynamicity of the web content, the use of anti-scraping mechanisms to detect and block automated scrapers, and the evolving nature of online threats. Some websites require user sessions and authentication to access content. Scraping such websites may involve simulating user interactions, including logging in. Websites frequently change their structure and layout, necessitating ongoing maintenance and updates to scraping scripts to ensure they continue to work correctly. Moreover, extracting webpage representative features from the web content may be inefficient for limited resources devices such as IoT devices. Although content-based features can be used for detecting many types of threats, relying on web content features is neither effective nor efficient for detecting advanced malicious websites.

The URL-based features seem to be a good alternative to the web content features. Many researchers compare the performance of the models constructed using both features and, on all occasions, URL-based features always win. However, most of the existing studies rely solely on the lexical features extracted from URLs. Lexical features have limited semantics information which causes the construction of sparse feature vectors. Some studies combine URL features with digital certificates to improve the detection performance. Malicious websites often lack valid certificates or use self-signed certificates, making certificate analysis a useful indicator of trustworthiness. Analyzing digital certificates can reveal whether a website is employing encryption, which is a common practice among reputable sites. However, not all websites use digital certificates, and some may employ self-signed certificates or certificates issued by less reputable Certificate Authorities (CAs). Extracting relevant and meaningful features from certificates for machine learning models can be complex, and the selection of the right features is crucial for effective detection. In addition, digital certificates can be misconfigured, expired, and frequently change leading to high false alarms. To sum up, existing solutions for detecting malicious web applications through web content analysis often struggle due to complex feature extraction, massive data volumes, evolving attack patterns, and limitations of traditional classifiers. Relying solely on lexical URL features proves insufficient, potentially leading to inaccurate classifications.

To address these challenges, this study proposes a novel multimodal representation approach that integrates textual and image-based features to enhance malicious website detection. This approach leverages the strengths of both modalities: textual features capture detailed semantic information related to attack patterns, and image features recognize broader malicious visual cues. Hidden patterns within textual content may become discernible through image analysis.

The proposed approach employs two Convolutional Neural Networks (CNNs): one for textual features and another for image features. Their outputs are then combined and fed into an artificial neural network classifier for improved decision-making. Our results demonstrate the superiority of the proposed model compared to existing approaches. We achieve a 4.3% increase in Matthews Correlation Coefficient (MCC) and a 1.5% reduction in the false-positive rate, showcasing the effectiveness of our multimodal approach in accurately identifying malicious web applications.

This study made the following contributions:

1. Integrating DNS-derived features with URL-based features enhances the comprehensiveness of malicious website detection. This synergy offers valuable contextual information regarding domain behavior and infrastructure, thereby fortifying the evaluation of website authenticity and security contributing to a more robust and nuanced approach to identifying malicious websites.
2. The study introduces a multimodal representation approach that utilizes both textual and image-based features to represent a comprehensive feature set. Textual features facilitate the deep learning model's ability to understand and represent detailed semantic information related to attack patterns, while image features are effective in recognizing more general malicious patterns.
3. Design and develop two Convolutional Neural Network (CNN) models to extract hidden features from the textual and image representations.
4. An additional, deep learning classifier was constructed to learn the relationships among the hidden features extracted by the CNN models. This approach advances the field by applying deep learning techniques to combine and leverage both textual and visual information for more effective malicious website detection.

The paper is organized as follows. Section 2 reviews the relevant literature and Section 3 describes the proposed solution in detail. Section 4 discusses the experimental design and Section 5 presents the results and discussion. Section 6 concludes the paper and discusses the limitations and future work.

II. RELATED WORK

There are three main approaches that have been suggested by researchers for malicious URL classification: blacklist, content-based, and URL-based [11, 32]. Many techniques were proposed to construct the detection classifiers such as the use of heuristic rules based on professional experience or the use of machine learning techniques. However, effective malicious URL detection is still an open issue problem. The performance of the recent malicious website detection solutions is influenced by the extracted features and the machine learning algorithms used for constructing the detection classifier. Authors in [32] presented an in-depth literature review that covers various machine learning-based techniques for detecting malicious URLs, considering aspects such as limitations, detection technologies, feature types, and datasets. The type of extracted features combined with deep learning techniques are research trends of malicious website detection solutions. The professional experience heuristic rule was widely used for constructing a blacklist of malicious URLs such as the Google safe web browsing tool [37]. However, the blacklist solutions are ineffective for malicious URL detection due to the constantly evolving threats causing the need for frequent identification of the evolved threat and frequently updating the database.

Many researchers have used feature extraction techniques to extract the features from website content to detect malicious content Natural language processing has been commonly employed for representation. However, due to the evolving nature of attacker’s techniques, malicious website content is complex and such patterns become dynamic and stealthy leading to poor detection accuracy. For example, in [38], the authors investigated how malicious websites employ various web spam techniques to evade detection. The aim is to provide an effective solution for detecting and combating malicious websites that utilize techniques like redirection spam, hidden Iframes spam, and content-hiding spam. Accordingly, the study focuses on capturing screenshots of webpages from a user's perspective and using a Convolutional Neural Network for classification. However, the solution is limited for detecting spam techniques. Moreover, the feature depends on screenshots of the loaded page might be dangerous and uncompleted due to the dynamic nature of the websites.

In [27], the authors collected features from the HTTP/s responses and applied various feature transformation and selection techniques for classification. However, these features are dynamic, subject to obfuscation using encoding and encryption mechanisms, which can render the detection classifier ineffective. Although machine learning algorithms were widely used for constructing the detection classifier, many researchers focused on deep learning techniques. Deep learning can accurately determine the similar patterns learned during the training resulting in effective classification. However, the web content is very dynamic and may be encrypted or encoded to hide the malicious patterns, posing a challenge in extracting effective features for classification.

The URL features which less dynamic are promising for the accurate detection of malicious domains. This is because malicious domains are generated algorithmically while benign domains are created by humans. Thus, malicious URLs may contain more prominent features compared to the features extracted from the content which can be obfuscated, or encrypted to mislead the learning process. Authors in [38] focused on detecting the malicious URLs that are generated algorithmically. They hypothesize that attackers or malicious bots are used to generate the malicious URLs automatically. Accordingly, those URLs may contain patterns that are different from those generated by humans. Similarly, authors in [39, 40] proposed solutions for detecting URLs that are generated using Domain Generation Algorithms (DGAs).

Authors in [41] proposed a malicious website detection technique based on lexical and host-based features extracted from URLs. Results showed that URL features are more accurate compared to the other types of features. Authors in [26] proposed an adaptive segmentation mechanism to solve the maximum sequence length (MSL) limitation in deep learning. Webpage text, digital certificate, and Uniform Resource Locator (URL) were used as the source of the extracted features and used to construct the detection model using the Multi-Head Self-Attention and multi-channel text convolution (MCTC) network. However, relying on dynamic content features is challenging and can lead to degrade the classification performance. The study in [42] presented an approach to learning the uncertainties by employing deep Bayesian neural networks (DBNNs) to model the stochastic system dynamics. Authors in [43] presented a feature extraction algorithm called URL embedding based unsupervised learning technique called Huffman coding to reduce the dimensionality of the features vector. Although the algorithm shows better detection performance compared to the existing feature extraction mechanisms, the algorithm has been evaluated using a dataset with a strong assumption about the length and distribution of the characters of the malicious URLs samples.

In [34], the authors proposed an anomaly detection model for detecting malicious domains. They utilized Hidden Markov Model (HMM) with a probabilistic model was used to construct the normal profile of the normal domain. In the online operation, if the domain is suspicious Jensen–Shannon divergence is calculated between the suspicious domain and a subset of the benign domains, and if the JS divergence exceeds a specific threshold the malicious domain is detected. Authors in [31] proposed a detection model called "deepBF" which combines Bloom Filters and Deep Learning techniques, aiming to improve accuracy and efficiency in identifying potentially harmful web addresses. The evolutionary convolutional neural network was used to construct the detection classifier. Authors in [33] compare the performance of several deep learning and traditional machine learning techniques to detect malicious URLs. The BiLSTM classifier was reported as the most performed classifier among studied classifiers.

Authors in [21] used a combination of different feature transformations to reduce the data volume to improve the learning process. Various linear and non-linear space transformation methods were used in the solution. Although feature transformation plays a significant role in improving the classifiers constructed using traditional machine learning techniques, the total number of features extracted is 62 features does not seem very challenging if deep learning techniques were used for the classification.

Authors in [44] presented a solution for malicious URL detection using two-stage ensemble learning to address the growing concern of web-based attacks. The study leverages cyber-threat intelligence features from sources like Google web search and Whois websites to enhance detection accuracy. The two-stage ensemble approach, combining Random Forest and Multi-Layer Perceptron algorithms, results in an improvement in accuracy and a reduction in false positives when compared to traditional URL-based models. However, the study does not thoroughly examine the potential limitations of relying on external cyber threat intelligence sources, which may pose challenges in terms of comprehensiveness and timeliness, warranting further investigation.

The authors in [45] proposed a curriculum-based multi-modal masked transformer network (CMMTN) that combines BERT and ResNet to enhance text and image representations, addressing the assumption of having labeled posts for training the fake news detection model. The CMMTN aims to strengthen correlations between relevant information by masking irrelevant context between modalities. However, the proposed solution in the current study is for malicious website detection, which presents different challenges compared to fake news detection, as it involves linguistic issues.

Authors in [46] introduced a multi-modal hierarchical attention model (MMHAM) for phishing website detection, extracting features from URLs, textual information, and visual design. However, the study solely focuses on phishing website detection, limiting its generalizability to other types of malicious websites. The current study takes a broader approach to detect various kinds of malicious websites. Additionally, it incorporates semantic textual patterns, utilizing Character embedding techniques to extract semantic features from textual data.

The authors in [47] proposed a hybrid deep learning approach to combine visual and textual modalities for detecting incongruous hashtags in user-generated content. However, the study concentrates on extracting contradictions between textual and visual features, which differs from malicious website detection where both features represent the same aspects from different perspectives.

To sum up, many approaches were investigated for detecting malicious websites and performance of detection relies heavily on the features extracted and the design of the model. Web content features are highly dynamic and complex, making it challenging to construct an efficient and effective classifier. For efficiency, the features should be rendered by a browsing machine before the extraction process which is risky and also needs valuable resources of memory and computational power for extracting the features. Meanwhile, for effectiveness, such features can be manipulated, encrypted, or encoded in such a way as to hide malicious patterns and make it very difficult to extract meaningful features for effective learning. URL features are more effective and efficient due to their size and generation conditions. The features extracted from URLs are less complex and more stable compared to the content-based features. Usually, malicious URLs are generated automatically using domain generation algorithms. Such URLs have different character distributions. That is the features can be more distinguishable compared to human-generated features. In addition, while features extracted from benign samples may be meaningful, malicious features usually contain meaningless terms, misspelled words, and randomly generated text. Benign URLs are more straightforward while malicious URLs may contain multiple domains, longer lengths, and contains more hercucal paths. Thus, features extracted from URLs contain more valuable patterns for the machine learning classifiers. Features such as those extracted from domain certificates or domain name servers are important. Lexical features extracted from domain information, URLs, and HTTP/s header response are also valuable. Features representation plays an essential role in improving learning performance. However, few studies focused on such issues. Many current detection models either rely on lexical features with statistical representations or depend on content-based features, which can result in low detection accuracy and high false alarms.

III. THE PROPOSED HF-CNN MODEL

The proposed model consists of four main phases as follows: features extraction phase, features representation phase, classifiers construction phase, and decision-making phase (See Figure 1). The output of each phase is used as input to the next phase. A detailed description of each phase is presented in the subsequent sections.

A. Phase 1: Data Collection Phase

The dataset used in this study is available on the Kagel website and can be downloaded from the following link (https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset?datasetId=1486586). Various types of URLs including benign, phishing, malware, and defacement, were collected from different sources such as the ISCX-URL-2016 dataset, Faizan's GitHub repository, and Malware Domain Blacklist dataset.

B. Phase 2: Features Extraction

Two types of features were extracted URLs-based and DNS-based features. The textual content presented in the URL is extracted using character-level n-gram to capture patterns, structures, and information present in the text of URLs. N-grams are contiguous sequences of n characters within the text. N-Gram is a text analysis technique that breaks down text into smaller units, where 'N' represents the number of units (typically words or characters). For example, in the URL "https://www.example.com," if we consider 3-grams (trigrams), we would have the following n-gram vector: ["htt", "ttp", "tps", "ps:", "s:/", "://", "//w", "/ww", "www", "ww.", "w.e", ".ex", "exa", "xam", "amp", "mpl", "ple", "le.c", "e.co", ".com"]. Each element in the n-gram vector is a feature. In this study n-gram that is ranged from 3 to 5 is used that is the features vector can contain complete textual terms such as “http”, “https”, “.com”, “.org” and so on. The DNS features are the information related to the DNS requests made when accessing these URLs. DNS requests may include domain names, IP addresses, and other metadata. Similar to the URL features DNS features were extracted and represented using n-gram.

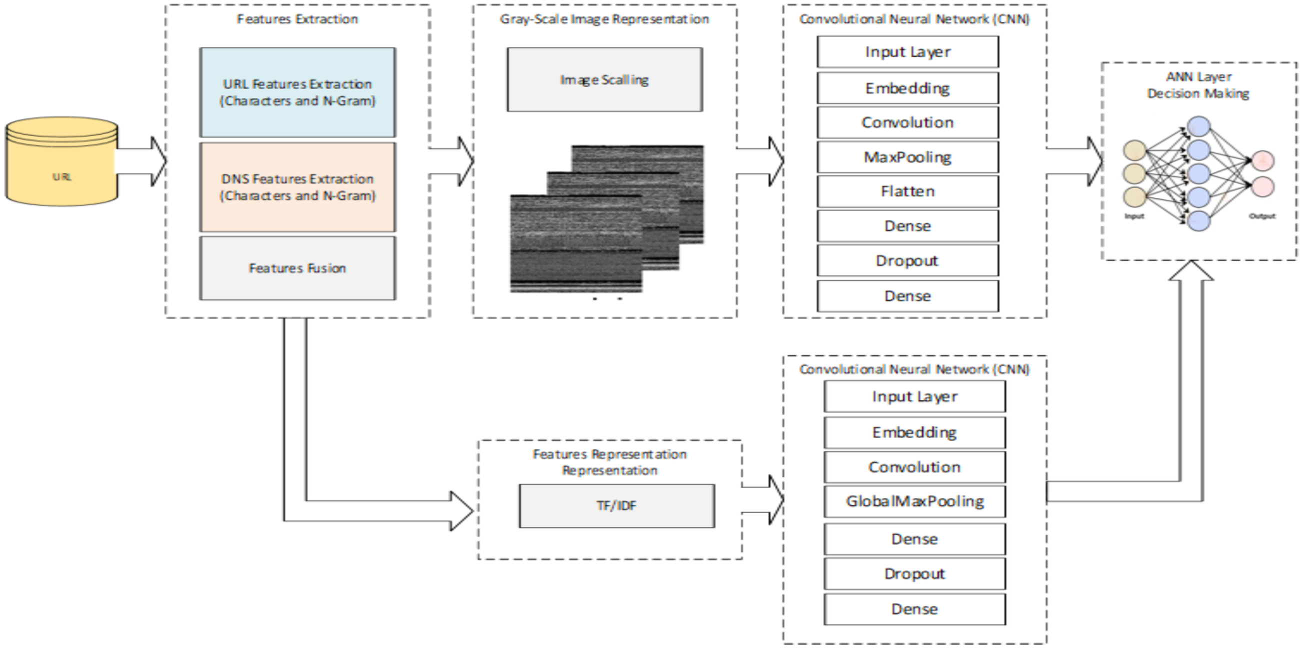

URL
Features Extraction
URL Features Extraction (Characters and N-Gram)
DNS Features Extraction (Characters and N-Gram)
Features Fusion
Gray-Scale Image Representation
Image Scalling
Convolutional Neural Network (CNN)
Input Layer
Embedding
Convolution
MaxPooling
Flatten
Dense
Dropout
Dense
ANN Layer
Decision Making
Input
Output
Features Representation Representation
TF/IDF
Convolutional Neural Network (CNN)
Input Layer
Embedding
Convolution
GlobalMaxPooling
Dense
Dropout
Dense

FIGURE 1. The Proposed HF-CNN Model for Malicious Website Detection

C. Phase 3: Features Representation

In this study, a multimodal representation approach employs textual and image-based features to represent the combined feature set. Textual features facilitate the deep learning model's ability to understand and represent detailed syntax information related to attack patterns, while image features are effective in recognizing more general malicious patterns.

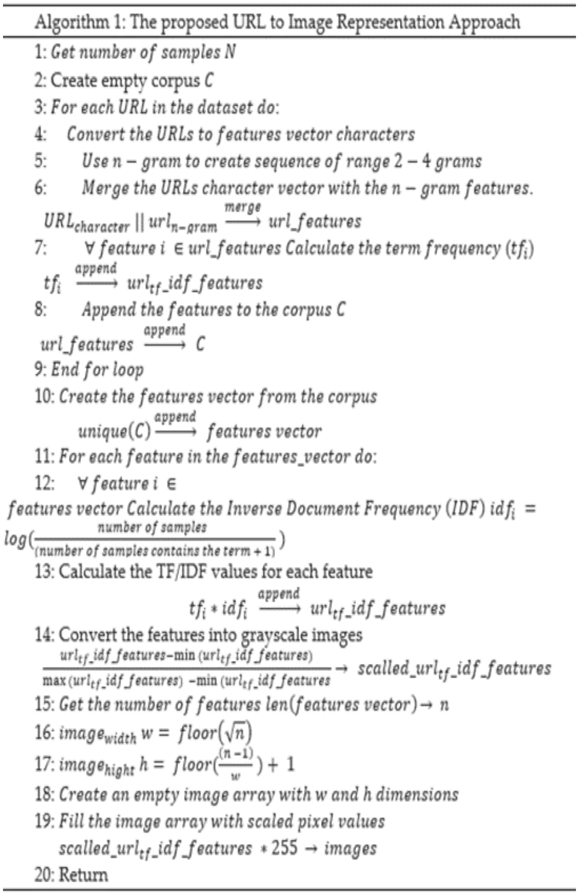
1) TEXT REPRESENTATION

The URLs are converted to sequences of characters called tokens. N-gram of range of (1,4) was used to enrich the features. Then a dictionary was created based on the unique tokens in the sequences. Then a feature vector containing all the unique tokens is constructed. For each token, an integer index is assigned. That is the dictionary that maps each token to a unique integer index. For example, if the word "www" is assigned index 3, that means it is the third token in order in the dictionary. The dictionary will also contain the frequency of the tokens in the entire corpus. Thus, to convert a URL to sequence the n-gram with a range of 1 to 4 is used to tokenize the URL at the character level, and then each token is mapped to it equaling count value in the dictionary. This sequence is post-padded based on the longest sequence in the dataset. For simplicity, the length of the sequence is set to 659 in this study. This sequence is used as input for the designed CNN input layer.

2) IMAGE REPRESENTATION:

URL information was treated as images. Each URL is converted into a visual representation, where characters in the URL are transformed into a 2D image-like structure. Character embedding was used. The resulting "images" represent the visual patterns within URLs. In this approach, each character in the URL is treated as a basic building block. The process of converting the URLs into visual images and converting them into a visual representation using character embedding consists of two steps. Firstly, the character-level Representation step in which the URL is broken down into its characters (letters, digits, symbols, etc.), and each character is considered as a discrete element. Secondly, in the features embedding step, Character embedding is a technique commonly used in Natural Language Processing (NLP) to represent discrete characters or words as continuous vectors. For each character in the URL, a corresponding embedding vector is generated. These vectors are learned during the training process and capture semantic information about the characters. Character embedding allows the model to convert characters into numerical representations that retain information about their relationships and patterns. The pseudo-code outlines the process of converting a URL into an image-like representation using character embedding and then using a CNN for feature extraction. Tokenize the URL into individual n-gram sequence

Let characters set is $C = \{abcdefghijklmnopqrstuvwxyz0123456789-,;.!?:'"/\backslash|\_@\#\$\%\hat{}\&*\tilde{}`+-=()[]\{\}\}$. The URL is converted to a series of characters. Each character is considered a feature. N-gram with a range between 2 to 4 was applied to extract more features from the URL to improve the representation. The n-gram features are merged into the URL character sets. Then, the term frequency tf_i is calculated for each feature in the merged vector. The term frequency of each feature is stored in a corpus called C (See algorithm 1 Line 8). The term frequency tf_i is a local measure of term importance within a single document. It gives you an idea of how often a word appears in a document. The unique terms in the corpus were extracted and stored in a dictionary. The inverse document frequency weight was calculated for each term in the dictionary as follows.



```
Algorithm 1: The proposed URL to Image Representation Approach
1: Get number of samples $N$
2: Create empty corpus $C$
3: For each URL in the dataset do:
4:     Convert the URLs to features vector characters
5:         Use $n-gram$ to create sequence of range $2-4$ grams
6:         Merge the URLs character vector with the $n-gram$ features.
   $URL_{character} \,||\, url_{n-gram} \xrightarrow{merge} url\_features$
7:         $\forall$ feature $i \in url\_features$ Calculate the term frequency $(tf_i)$
   $tf_i \xrightarrow{append} url_{tf}\_idf\_features$
8:         Append the features to the corpus $C$
   $url\_features \xrightarrow{append} C$
9: End for loop
10: Create the features vector from the corpus
        $unique(C) \xrightarrow{append}$ features vector
11: For each feature in the features_vector do:
12:     $\forall$ feature $i \in$
features vector Calculate the Inverse Document Frequency (IDF) $idf_i = log(\frac{number\ of\ samples}{(number\ of\ samples\ contains\ the\ term + 1)})$
13: Calculate the TF/IDF values for each feature
        $tf_i * idf_i \xrightarrow{append} url_{tf}\_idf\_features$
14: Convert the features into grayscale images
   $\frac{url_{tf}\_idf\_features - \min(url_{tf}\_idf\_features)}{\max(url_{tf}\_idf\_features) - \min(url_{tf}\_idf\_features} \rightarrow scalled\_url_{tf}\_idf\_features$
15: Get the number of features $len(features\ vector) \rightarrow n$
16: $image_{width}\ w = floor(\sqrt{n})$
17: $image_{hight}\ h = floor(\frac{(n-1)}{w}) + 1$
18: Create an empty image array with $w$ and $h$ dimensions
19: Fill the image array with scaled pixel values
      $scalled\_url_{tf}\_idf\_features * 255 \rightarrow images$
20: Return
```

$$idf_i = log\left(\frac{number\ of\ samples}{(number\ of\ samples\ contains\ the\ term + 1)}\right) \quad (1)$$

where the idf_i is the document frequency. IDF measures the global importance of a term across the entire corpus by multiplying the $tf_i$and idf_ivalues for each term in each document. This results in a TF-IDF score for each term in each document. It quantifies how unique or common a term is in the corpus. Next, for each feature in the corpus, the term frequency-inverse term frequency (tf_idf) is calculated as follows.

$$t\_idf_i = tf_i * idf_i \quad (2)$$

The t_idf_i score for a term in a document is higher if the term appears frequently in that document but is relatively rare across the entire corpus. The t_idf_i features are scaled using min-max normalization as follows.

$$scalled\_url_{tf}\_idf\_features = \frac{url_{tf}\_idf\_features - min\ (url_{tf}\_idf\_features)}{max\ (url_{tf}\_idf\_features)\ - min\ (url_{tf}\_idf\_features} \quad (3)$$

Finally, the features vector is created from the unique terms of the corpus. The maximum length of the feature vector is 4096 features. These features vector was converted to 64×64 image size as follows.

$$image_{width}\ w = floor(\sqrt{n}) \qquad (4)$$

$$image_{hight}\ h = floor(\frac{(n\ -1)}{w}) + 1 \qquad (5)$$

The pseudocode in Algorithm 1 illustrates the proposed URL to image representation approach and Figure 2 shows the output of the algorithm. Figure 3 shows the histogram of six samples selected randomly. As can be seen in Figures 2 and 3 benign websites have less intense features compared to defacement websites. Phishing websites look similar to benign websites it can be interpreted by the attackers' purpose. In phishing websites, attackers try to look benign so they can harvest sensitive information or perform an attack.

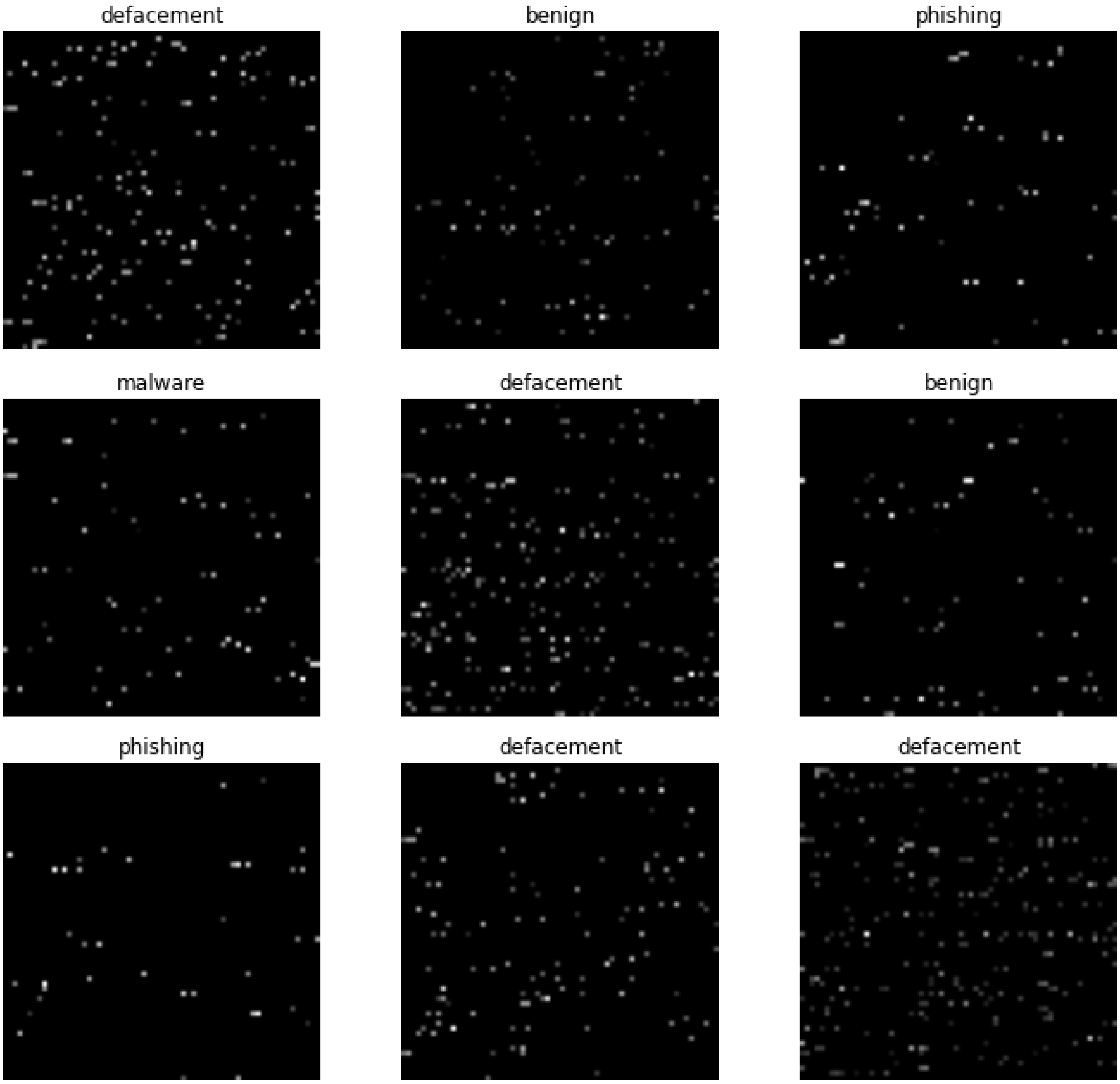


FIGURE 2. **The output of the proposed algorithm URLs to image**

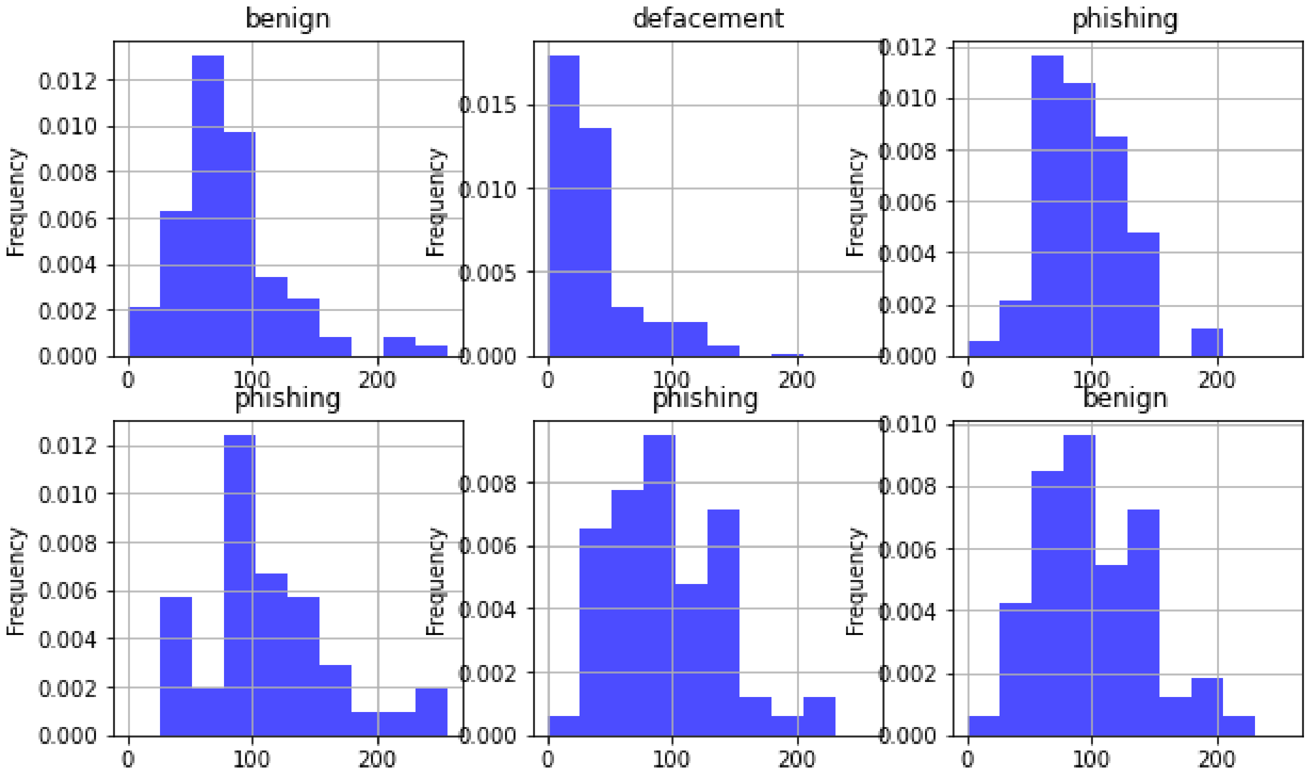


FIGURE 3. **The histogram of six selected image samples.**

D. Phase 4: CNN Models Construction

Two CNN models were constructed the first model was trained based on the image representation features and the other based on the textual-based features. The detailed description of these two models is presented as follows.

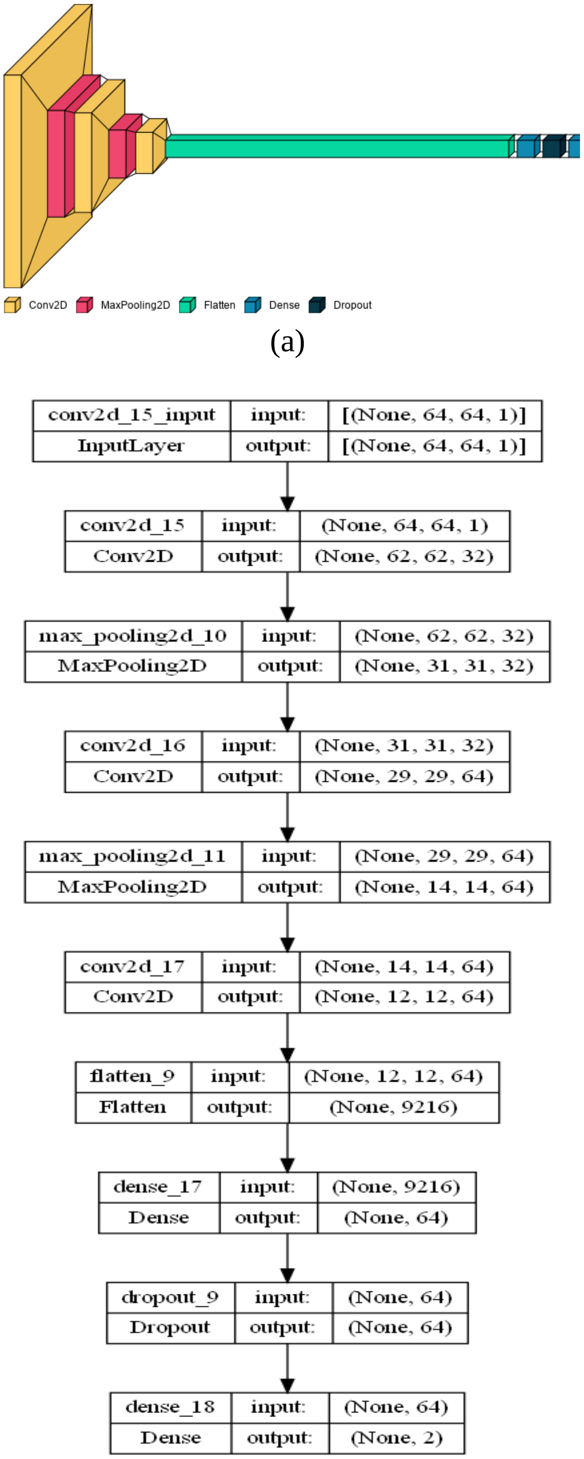
1) CNN MODEL FOR IMAGE:

CNNs are typically used for image-related tasks, as they are effective at detecting patterns and features in 2D data. By applying convolutional layers to the grid of the images represented by the proposed Algorithm 1, CNN learns to detect important patterns and features within the URL's character sequence. As shown in Figures 4(a) and (b), the proposed CNN model, which is called imgCNN consists of nine layers as follows.

The first layer is the convolutional layer with 32 filters/kernels, a kernel size of (3, 3), and ReLU activation. It processes the input data, resulting in feature maps of size (62, 62, 32). The second layer is the max-pooling layer with a pool size of (2, 2). It reduces the spatial dimensions of the feature maps by taking the maximum value in each 2x2 region, resulting in smaller feature maps. The Output Shape of this layer is 31 × 31 size images (None, 31, 31, 32). The third layer is the second convolutional layer with 64 filters, a kernel size of (3, 3), and ReLU activation. It further processes the feature maps from the previous layer. The output shape of this layer is (None, 29, 29, 64). The fourth layer is the second max-pooling layer with a pool size of (2, 2), further reducing the spatial dimensions. The fifth layer is the third convolutional layer with 64 filters, a kernel size of (3, 3), and ReLU activation. The sixth layer flattens the 3D feature maps into a 1D vector, preparing them for fully connected layers. The seventh layer is a fully connected layer which has 64 units with ReLU activation.

2) CNN MODEL FOR TEXTUAL FEATURES

As shown in Figure 5, the proposed deep learning model for malicious URL classification using text representation (txtCNN) relies on a 1D Convolutional Neural Network (CNN). It commences with an embedding layer that translates the character-level inputs with n-gram features into continuous 32-dimensional vectors. Following this, a 1D convolutional layer of 128 filters and ReLU activation is applied to capture salient features in the text data. Max-pooling is subsequently employed for spatial reduction. The flattened output is then processed through a dense layer consisting of 128 units with ReLU activation. To mitigate overfitting, dropout with a rate of 0.5 is introduced. Finally, the model employs a softmax-based output layer to provide classification probabilities for the defined number of classes. This architecture excels at learning meaningful patterns in textual representations of URLs, facilitating the distinction between benign and malicious URLs.


Conv2D
MaxPooling2D
Flatten
Dense
Dropout
conv2d_15_input | InputLayer | input: [(None, 64, 64, 1)] | output: [(None, 64, 64, 1)]
conv2d_15 | Conv2D | input: (None, 64, 64, 1) | output: (None, 62, 62, 32)
max_pooling2d_10 | MaxPooling2D | input: (None, 62, 62, 32) | output: (None, 31, 31, 32)
conv2d_16 | Conv2D | input: (None, 31, 31, 32) | output: (None, 29, 29, 64)
max_pooling2d_11 | MaxPooling2D | input: (None, 29, 29, 64) | output: (None, 14, 14, 64)
conv2d_17 | Conv2D | input: (None, 14, 14, 64) | output: (None, 12, 12, 64)
flatten_9 | Flatten | input: (None, 12, 12, 64) | output: (None, 9216)
dense_17 | Dense | input: (None, 9216) | output: (None, 64)
dropout_9 | Dropout | input: (None, 64) | output: (None, 64)
dense_18 | Dense | input: (None, 64) | output: (None, 2)

(a)

(b)

FIGURE 4. The structure of the proposed imgCNN Model

As the URL representation passes through the CNN, the network performs feature extraction. Features might include detecting specific character combinations, sequences, or other visual patterns within the URL. The CNN learns to recognize which patterns are indicative of certain URL categories, such as malicious or benign. The output from the CNN is then used as a feature representation of the URL. This feature representation, which captures visual patterns within the URL, can be passed to further layers in the neural network for classification.

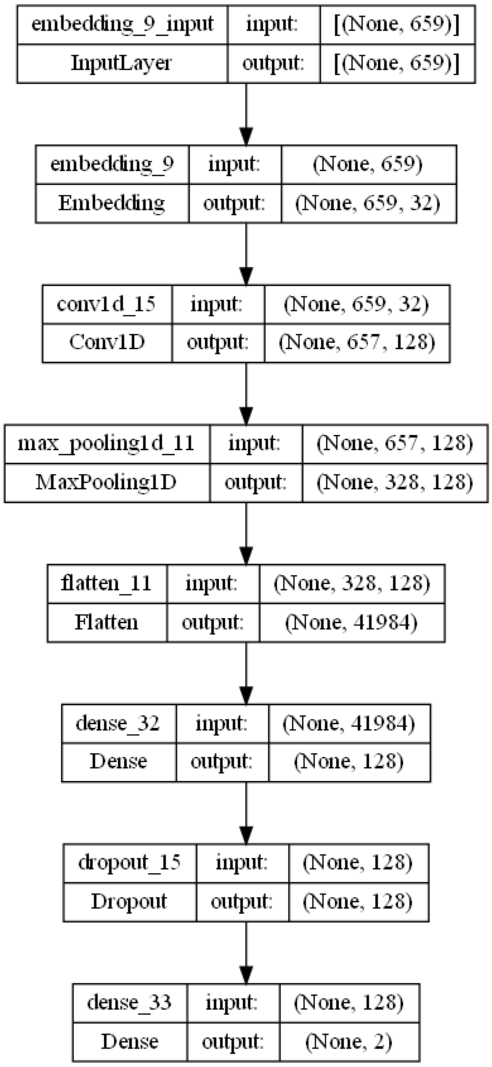

embedding_9_input | InputLayer | input: [(None, 659)] | output: [(None, 659)]
embedding_9 | Embedding | input: (None, 659) | output: (None, 659, 32)
conv1d_15 | Conv1D | input: (None, 659, 32) | output: (None, 657, 128)
max_pooling1d_11 | MaxPooling1D | input: (None, 657, 128) | output: (None, 328, 128)
flatten_11 | Flatten | input: (None, 328, 128) | output: (None, 41984)
dense_32 | Dense | input: (None, 41984) | output: (None, 128)
dropout_15 | Dropout | input: (None, 128) | output: (None, 128)
dense_33 | Dense | input: (None, 128) | output: (None, 2)

FIGURE 5. The structure of the proposed txtCNN Model

E. Phase 5: Decision Making

The decision-making model is a sequential deep learning model designed to classify URLs as either benign or malicious based on integrated features from two separate models, one processing URL text representations and the other treating URLs as images. As shown in Figure 6, the model begins with an input layer, followed by densely connected layers with ReLU activation functions. These layers collectively enable the model to learn complex patterns and representations from both text and image data. The final output layer employs the softmax activation function to provide class probabilities for classification. The model is optimized using the Adam optimizer and trained to minimize categorical cross-entropy loss. Its architecture allows it to effectively fuse information from text and image representations, making informed decisions about the nature of URLs, and contributing to robust URL classification.

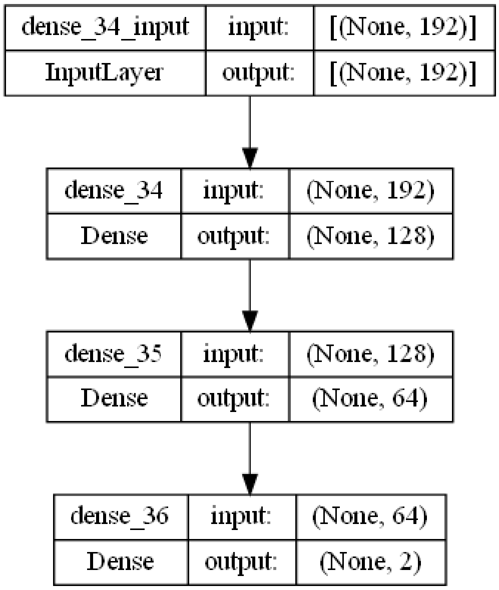


FIGURE 6. The structure of the decision-making HF-CNN Model

IV. PERFORMANCE EVALUATION

The dataset, the experimental procedures, and the performance evaluation are described in the following sub-sections.

A. Sources and Preprocessing of Datasets

In this study, a popular and accessible dataset of malicious URLs was used. This dataset can be found on the Kaggle.com repository [48]. The dataset was sourced from well-established repositories frequently used by researchers specializing in the detection of malicious URLs, including Phishtank [39,40] (accessible at https://phishtank.org/) and the URL dataset known as ISCX-URL-2016 [8] (available at https://www.unb.ca/cic/datasets/url-2016.html). The URLs within this dataset are either malicious or benign. The malicious URLs encompassed a range of types, such as links to malware, web defacement, spam, phishing, and drive-by downloads. In this study, a sample of 50,000 URLs was randomly selected. Because some URLs are outdated, the validity of the URLs was tested before it is included in the sample dataset. An http/s request was initiated for each URL in the dataset, only the valid HTTP response was included in the sample dataset. Figure 7 presents a summary of the quantity and various types of URL samples present in the original dataset (right figure) and the selected sample (left figure).

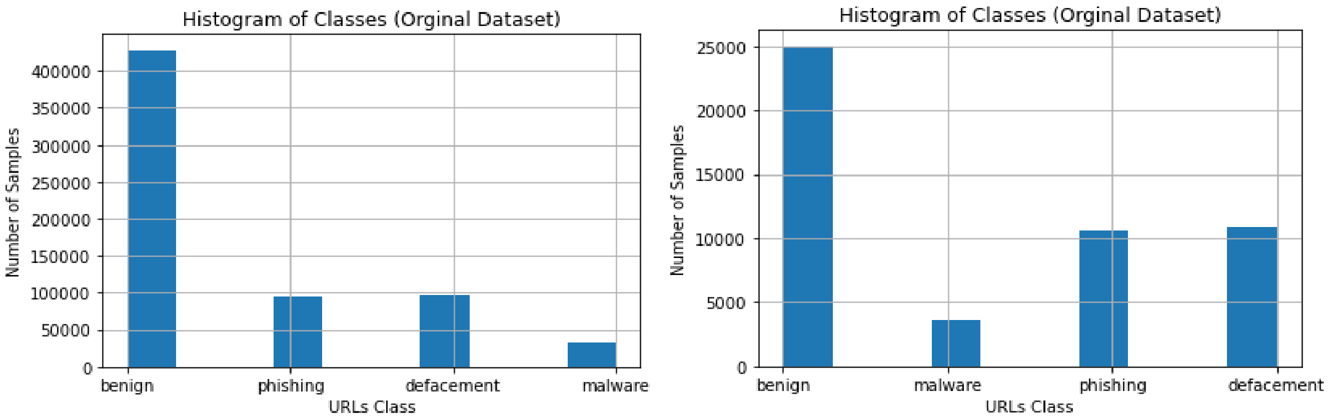


Figure 7. Classes Histogram: (right) Original Dataset (left) Sample Dataset

B. Experimental Procedures

In this study, the state-of-the-art deep learning-based solutions, which have previously been proposed for malicious URL detection, were used for the evaluation of the proposed model. Additionally, text-based CNN and Image-based CNN were developed to serve as baselines for evaluating the proposed model. The lexical URL-based features, drawing from existing literature [6, 9, 11-13, 18, 49] were also used in the comparison. In the subsequent section, we provide a detailed exposition of the results.

4.3. Performance Measure

To assess the detection performance of the proposed model, we employed five key performance metrics: overall accuracy, detection rate (recall), precision, F1 score, Matthews Correlation Coefficient (MCC), false-positive rate (FPR), and false-negative rate (FNR). These performance metrics are widely accepted and commonly utilized in the evaluation of malware detection solutions within the existing literature. The MCC measures the quality of binary classifications, particularly when dealing with imbalanced datasets. It takes into account true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) to provide a balanced evaluation of a binary classification model. The performance measures used in this study were calculated based on the following equations.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

$$FPR = \frac{FP}{TP + FN} \quad (7)$$

$$FNR = \frac{FN}{TN + FP} \quad (7)$$

$$DR\ (Recall) = \frac{TP}{TP + FN} \quad (8)$$

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

$$\mathrm{F-measure} = \frac{2 \times Precision\ \times Recall}{Precision + Recall} \quad (10)$$

$$\mathrm{MCC} = \frac{TP \times TN - FP\ \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (11)$$

Although the F-measure evaluates the overall performance of the model by measuring the balance between precision and recall, it doesn't consider true negatives, making it less informative for imbalanced datasets. The MCC is a more accurate measure because it is sensitive to class distribution and dataset size. MCC takes into account both true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) in a balanced way. Therefore, it gives more insights into the performance of the model.

V. RESULTS AND DISCUSSION

The classification results of the proposed HF-CNN and imgCNN as compared to the related work models are listed in Table 1. It can be seen that the proposed HF-CNN is superior to all other studied models. Compared with the baseline model txtCNN, the proposed HF-CNN is 0.7%, 0.7%, 0.4%, and 0.6% improvement in terms of Accuracy, Precession, Recall, F-Measure, and MCC, respectively. The False Positive Rate (FPR) and False Negative Rate (FNR) were reduced by 1.6% and 1.4%, respectively.

Figures 7-13 present results of the proposed HF-CNN, and imgCNN as compared to the related work models, in terms of Accuracy, Precession, Recall, F-Measure, MCC, FNR, and FPR respectively. As can be seen in these figures, CNN models outperform the other studied models. LSTM and DBN achieved lower performance compared to the other studied model this is because LSTM and DBN models are designed for sequence modeling where there are clear dependencies between elements in a sequence. Malicious URL patterns, however, may not exhibit strong sequential dependencies, making LSTM and DBN less effective for URL classification. BiLSTM, however, achieved better performance than the LSTM. The LSTM is likely unable to capture the spatial correlation among the URL features while BiLSTMs, with their bidirectional processing, can capture spatial context features. MCCNN and AMCCNN achieved comparable good performance compared with the proposed model (See Figures 7 and 8). Both MCCNN and AMCCNN models employ CNN to extract and classify the URLs. CNN-based models can capture the spatial dependencies in the URL features. This interprets also the improvement gained when the URLs are represented as images and the CNN model is used for classification. CNNs are designed for processing grid-like data, such as images, which have a clear spatial structure. CNNs are capable of capturing both local features (e.g., character-level patterns) and global features (e.g., overall URL structure) simultaneously. This flexibility allows them to identify malicious patterns at different scales within URLs.

TABLE I
Performance Evaluation

Model	Accuracy	Precision	Recall	F-Measure	MCC	FNR	FPR
HF-CNN	**98.51%**	**98.25%**	**99.52%**	**98.88%**	**96.66%**	**0.48%**	**3.49%**
imgCNN	98.33%	98.20%	99.28%	98.73%	96.28%	0.72%	3.49%
txtCNN	97.77%	97.51%	99.14%	98.32%	95.05%	0.86%	4.84%
DBN	90.75%	88.99%	98.03%	93.29%	79.46%	1.97%	23.14%
LSTM	87.85%	86.59%	96.87%	91.44%	72.07%	3.13%	30.45%
BiLSTM	96.60%	97.27%	97.64%	97.46%	92.34%	2.36%	5.48%
MCCNN	96.68%	97.11%	98.01%	97.56%	92.36%	1.99%	6.12%
AMCCNN	96.43%	97.71%	96.85%	97.28%	92.08%	3.15%	4.41%
LR	94.13%	94.37%	96.79%	95.56%	86.94%	3.21%	10.89%
RF	96.20%	96.10%	98.16%	97.12%	91.57%	1.84%	7.50%
SVM	95.48%	95.01%	98.24%	96.60%	89.97%	1.76%	9.74%

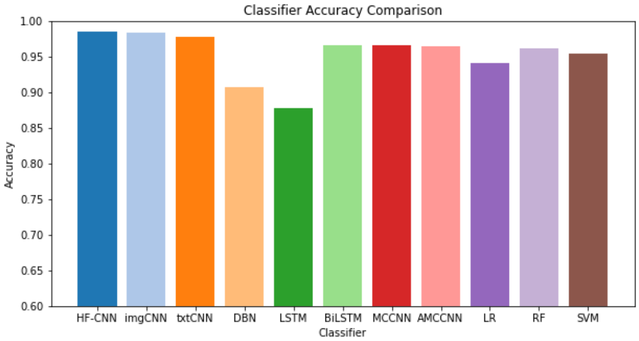


FIGURE 7. **Figure 4. Comparison in terms of the Accuracy**

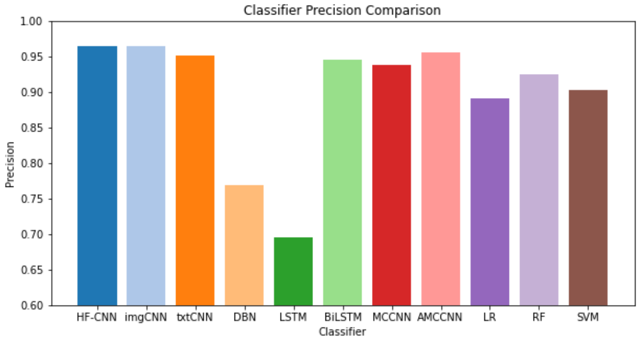


FIGURE 8. **Figure 5. Comparison in terms of the Precession**

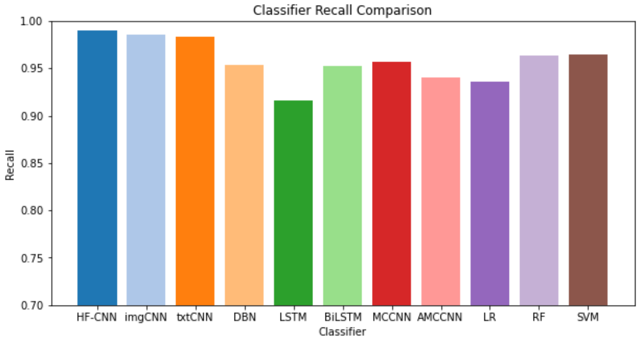


FIGURE 9. **Figure 6. Comparison in terms of the Recall**

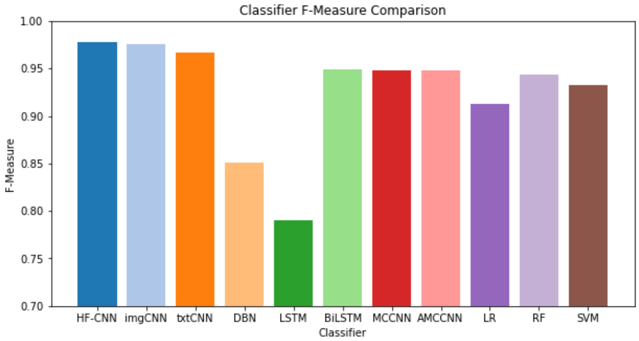


FIGURE 10. **Comparison in terms of the F-Measure**

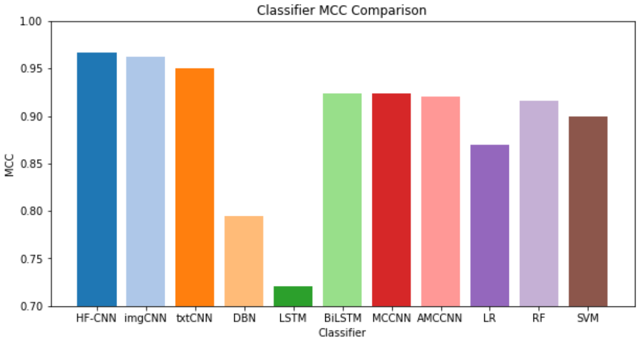


FIGURE 11. **Comparison in terms of the MCC**

Figure 12 and 13 shows the results in terms of false positive rate (FPR) and false negative rate (FNR). Both measures are important in the evaluation of the malicious website detection models. As can be noticed in Figure 9 the proposed models HF-CNN and imgCNN achieved the lowest false positive rate which is 3.49% for both models (Seet Table 1). The DBN and LSTM models achieved 23.14%, and 30.45% respectively. CNN models are more effective in eliminating the false positive rate, due to their ability to capture the malicious pattern in the URLs features. Traditional machine learning produced a high rate of false positives because such algorithms do not capture complex sequential or spatial dependencies present in the URL-based features. Although most of the models achieved a false negative rate lower than 3%, however, such a percentage could be dangerous for critical systems. Recent studies show that an average US internet user visits 130 web pages per day. That is, every day an average internet user may visit 39 malicious websites per thousand URLs. The proposed model achieved a 0.48% of the false negative rate. That is 6.24 malicious websites might be visited per each thousand visited URLs.

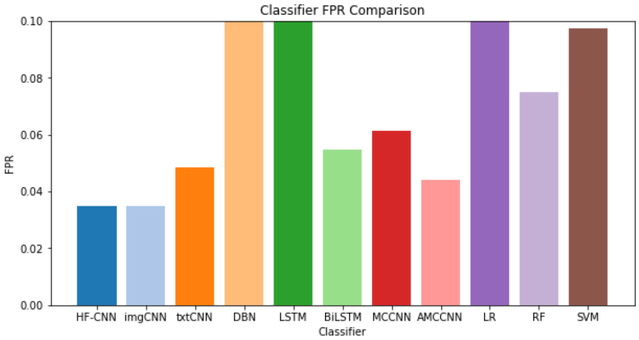


FIGURE 12. **Comparison in terms of the FPR**

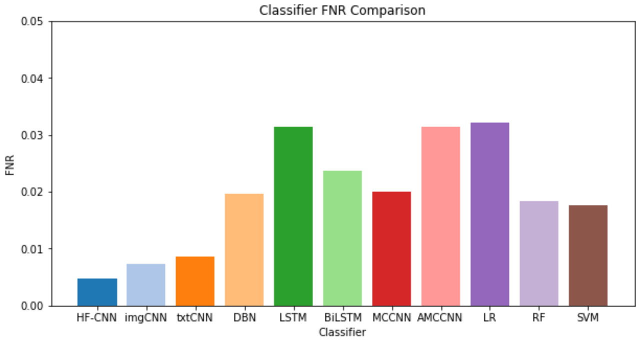


FIGURE 13. **Comparison in terms of the FNR**

The results exhibited that the URL-based features are promising alternatives to web content features. Researchers often assess model performance by comparing both sets of features, and consistently, URL-based features outperform their counterparts. Nevertheless, the majority of existing studies primarily rely on lexical features extracted from URLs, which offer limited semantic information and result in sparse feature vectors. Some studies seek to enhance detection performance by combining URL features with digital certificates. Malicious websites frequently lack valid certificates or resort to self-signed certificates, rendering certificate analysis a valuable trustworthiness indicator. Evaluating digital certificates can unveil whether a website employs encryption, a common practice among reputable sites. However, not all websites employ digital certificates, and some may utilize self-signed certificates or certificates issued by less reputable Certificate Authorities (CAs). The extraction of relevant and meaningful features from certificates for machine learning models can be intricate, and the judicious selection of appropriate features is pivotal for effective detection. Furthermore, digital certificates can be susceptible to misconfiguration, expiration, and frequent changes, leading to an elevated rate of false alarms.

VI. CONCLUSIONS AND FUTURE WORKS

In this study, a malicious website detection model called HF-CNN was designed and developed. The model integrates URL features with DNS features to enhance the comprehensiveness of identifying malicious websites. A multimodal representation approach that encompasses both textual and image-based characteristics has been proposed to depict the combined feature set. Textual attributes enable the deep learning model to grasp and depict complex semantic details associated with attack patterns, while image attributes surpass at recognizing broader malicious patterns. Two Convolutional Neural Network (CNN) models were constructed to extract hidden features from the textual and image representations. CNNs are capable of simultaneously capturing both local and global features. The results indicate that the proposed model outperforms the other related models. The overall performance in terms of F-measure and MCC has been improved by 0.4%, and 0.6%, respectively, compared with the baseline model txtCNN. The False Positive Rate (FPR) and False Negative Rate (FNR) were reduced by 1.6% and 1.4%, respectively.

While the proposed models achieved a high detection performance of 98.88% in terms of F-measure, there are still considerable amounts of errors presented in the detection performance as measured by the MMC score of 96.66%. The errors mostly resulted from the unrepresented features in URLs and DNS information. Therefore, relying solely on URLs, DNS information or static features is not a wise approach to malicious website detection, as some benign domains that suffer from security vulnerabilities may become malicious due to injection attacks. Therefore, it is important to combine the URL-based features with other features such as content features. However, content features are complex due to their high dynamicity and usability by attackers to evade detection. As a result, further research is needed to propose effective and efficient mechanisms for acquiring web content. Furthermore, employing an adaptive ensemble of classifiers designed to accommodate the dynamic nature of evolving threats could enhance detection performance. Each classifier within the ensemble is constructed based on a distinct set of features, providing versatility and robustness in addressing diverse threat scenarios.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia, for funding this research work; project number (168/442). Further, the authors would like to extend their appreciation to Taibah University for its supervision support.

REFERENCES

[1] NJ. "How Many Websites Are There in the World?" https://siteefy.com/how-many-websites-are-there/ (accessed 10/9/2023, 2023).

[2] M. Liu, B. Zhang, W. Chen, and X. Zhang, "A survey of exploitation and detection methods of XSS vulnerabilities," IEEE Access, vol. 7, pp. 182004-182016, 2019.

[3] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," Journal of Computer and System Sciences, vol. 80, no. 5, pp. 973-993, 2014/08/01/ 2014, doi: https://doi.org/10.1016/j.jcss.2014.02.005.

[4] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu, "The Ghost in the Browser: Analysis of Web-based Malware," HotBots, vol. 7, pp. 4-4, 2007.

[5] K. Townsend. "18.5 Million Websites Infected With Malware at Any Time." Wired Business Media. https://www.securityweek.com/185-million-websites-infected-malware-any-time (accessed 1/2/2022, 2022).

[6] A. Saleem Raja, R. Vinodini, and A. Kavitha, "Lexical features based malicious URL detection using machine learning techniques," Materials Today: Proceedings, vol. 47, pp. 163-166, 2021/01/01/ 2021, doi: https://doi.org/10.1016/j.matpr.2021.04.041.

[7] A. Subasi, M. Balfaqih, Z. Balfagih, and K. Alfawwaz, "A Comparative Evaluation of Ensemble Classifiers for Malicious Webpage Detection," Procedia Computer Science, vol. 194, pp. 272-279, 2021/01/01/ 2021, doi: https://doi.org/10.1016/j.procs.2021.10.082.

[8] S. Rameem Zahra, M. Ahsan Chishti, A. Iqbal Baba, and F. Wu, "Detecting Covid-19 chaos driven phishing/malicious URL attacks by a fuzzy logic and data mining based intelligence system," Egyptian

Informatics Journal, 2021/12/14/ 2021, doi: https://doi.org/10.1016/j.eij.2021.12.003.

[9] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment," Computer Communications, vol. 175, pp. 47-57, 2021/07/01/ 2021, doi: https://doi.org/10.1016/j.comcom.2021.04.023.

[10] R. Wazirali, R. Ahmad, and A. A.-K. Abu-Ein, "Sustaining accurate detection of phishing URLs using SDN and feature selection approaches," Computer Networks, vol. 201, p. 108591, 2021/12/24/ 2021, doi: https://doi.org/10.1016/j.comnet.2021.108591.

[11] D. K. Mondal, B. C. Singh, H. Hu, S. Biswas, Z. Alom, and M. A. Azim, "SeizeMaliciousURL: A novel learning approach to detect malicious URLs," Journal of Information Security and Applications, vol. 62, p. 102967, 2021/11/01/ 2021, doi: https://doi.org/10.1016/j.jisa.2021.102967.

[12] K. Haynes, H. Shirazi, and I. Ray, "Lightweight URL-based phishing detection using natural language processing transformers for mobile devices," Procedia Computer Science, vol. 191, pp. 127-134, 2021/01/01/ 2021, doi: https://doi.org/10.1016/j.procs.2021.07.040.

[13] S. Srinivasan, R. Vinayakumar, A. Arunachalam, M. Alazab, and K. Soman, "DURLD: Malicious URL Detection Using Deep Learning-Based Character Level Representations," in Malware Analysis Using Artificial Intelligence and Deep Learning: Springer, 2021, pp. 535-554.

[14] R. Chiramdasu, G. Srivastava, S. Bhattacharya, P. K. Reddy, and T. R. Gadekallu, "Malicious URL Detection using Logistic Regression," in 2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS), 23-25 Aug. 2021 2021, pp. 1-6, doi: 10.1109/COINS51742.2021.9524269.

[15] N. M. Phung and M. Mimura, "Detection of malicious javascript on an imbalanced dataset," Internet of Things, vol. 13, p. 100357, 2021/03/01/ 2021, doi: https://doi.org/10.1016/j.iot.2021.100357.

[16] Y. Huang, T. Li, L. Zhang, B. Li, and X. Liu, "JSContana: Malicious JavaScript detection using adaptable context analysis and key feature extraction," Computers & Security, vol. 104, p. 102218, 2021/05/01/ 2021, doi: https://doi.org/10.1016/j.cose.2021.102218.

[17] R. Rakesh, S. Muthurajkumar, L. SaiRamesh, M. Vijayalakshmi, and A. Kannan, "Detection of URL based attacks using reduced feature set and modified C4. 5 algorithm," Advances in Natural and Applied Sciences, vol. 9, no. 6 SE, pp. 304-311, 2015.

[18] S. Kim, J. Kim, and B. B. Kang, "Malicious URL protection based on attackers' habitual behavioral analysis," Computers & Security, vol. 77, pp. 790-806, 2018/08/01/ 2018, doi: https://doi.org/10.1016/j.cose.2018.01.013.

[19] S. He, B. Li, H. Peng, J. Xin, and E. Zhang, "An Effective Cost-Sensitive XGBoost Method for Malicious URLs Detection in Imbalanced Dataset," IEEE Access, vol. 9, pp. 93089-93096, 2021.

[20] D. R. Patil and J. B. Patil, "Malicious URLs detection using decision tree classifiers and majority voting technique," Cybernetics and Information Technologies, vol. 18, no. 1, pp. 11-29, 2018.

[21] T. Li, G. Kou, and Y. Peng, "Improving malicious URLs detection via feature engineering: Linear and nonlinear space transformation methods," Information Systems, vol. 91, p. 101494, 2020/07/01/ 2020, doi: https://doi.org/10.1016/j.is.2020.101494.

[22] S. Wang et al., "Deep and broad URL feature mining for android malware detection," Information Sciences, vol. 513, pp. 600-613, 2020/03/01/ 2020, doi: https://doi.org/10.1016/j.ins.2019.11.008.

[23] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Evaluating deep learning approaches to characterize and classify malicious URL's," Journal of Intelligent & Fuzzy Systems, vol. 34, pp. 1333-1343, 2018, doi: 10.3233/JIFS-169429.

[24] B. Liang, M. Su, W. You, W. Shi, and G. Yang, "Cracking classifiers for evasion: A case study on the google's phishing pages filter," in Proceedings of the 25th International Conference on World Wide Web, 2016, pp. 345-356.

[25] Y. Shi, G. Chen, and J. Li, "Malicious domain name detection based on extreme machine learning," Neural Processing Letters, vol. 48, pp. 1347-1357, 2018.

[26] G. Sun, Z. Zhang, Y. Cheng, and T. Chai, "Adaptive segmented webpage text based malicious website detection," Computer Networks, vol. 216, p. 109236, 2022/10/24/ 2022, doi: https://doi.org/10.1016/j.comnet.2022.109236.

[27] J. McGahagan IV, D. Bhansali, C. Pinto-Coelho, and M. Cukier, "Discovering Features for Detecting Malicious Websites: An Empirical Study," Computers & Security, p. 102374, 2021.

[28] N. Samarasinghe and M. Mannan, "On cloaking behaviors of malicious websites," Computers & Security, vol. 101, p. 102114, 2021/02/01/ 2021, doi: https://doi.org/10.1016/j.cose.2020.102114.

[29] S. Kim, J. Kim, S. Nam, and D. Kim, "WebMon: ML- and YARA-based malicious webpage detection," Computer Networks, vol. 137, pp. 119-131, 2018/06/04/ 2018, doi: https://doi.org/10.1016/j.comnet.2018.03.006.

[30] J. McGahagan, D. Bhansali, C. Pinto-Coelho, and M. Cukier, "Discovering features for detecting malicious websites: An empirical study," Computers & Security, vol. 109, p. 102374, 2021/10/01/ 2021, doi: https://doi.org/10.1016/j.cose.2021.102374.

[31] R. Patgiri, A. Biswas, and S. Nayak, "deepBF: Malicious URL detection using learned Bloom Filter and evolutionary deep learning," Computer Communications, vol. 200, pp. 30-41, 2023/02/15/ 2023, doi: https://doi.org/10.1016/j.comcom.2022.12.027.

[32] M. Aljabri et al., "Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions," IEEE Access, vol. 10, pp. 121395-121417, 2022, doi: 10.1109/ACCESS.2022.3222307.

[33] V. Devalla, S. Srinivasa Raghavan, S. Maste, J. D. Kotian, and D. D. Annapurna, "mURLi: A Tool for Detection of Malicious URLs and Injection Attacks," Procedia Computer Science, vol. 215, pp. 662-676, 2022/01/01/ 2022, doi: https://doi.org/10.1016/j.procs.2022.12.068.

[34] H. Wang, Z. Tang, H. Li, J. Zhang, and C. Cai, "DDOFM: Dynamic malicious domain detection method based on feature mining," Computers & Security, vol. 130, p. 103260, 2023/07/01/ 2023, doi: https://doi.org/10.1016/j.cose.2023.103260.

[35] G. Palaniappan, S. S, B. Rajendran, Sanjay, S. Goyal, and B. B S, "Malicious Domain Detection Using Machine Learning On Domain Name Features, Host-Based Features and Web-Based Features," Procedia Computer Science, vol. 171, pp. 654-661, 2020/01/01/ 2020, doi: https://doi.org/10.1016/j.procs.2020.04.071.

[36] M. A. Khan, M. M. Nasralla, M. M. Umar, R. Ghani Ur, S. Khan, and N. Choudhury, "An Efficient Multilevel Probabilistic Model for Abnormal Traffic Detection in Wireless Sensor Networks," Sensors, vol. 22, no. 2, 2022, doi: 10.3390/s22020410.

[37] P. K. Sandhu and S. Singla, "Google safe browsing-web security," IJCSET, vol. 5, no. 7, pp. 283-287, 2015.

[38] S. Yadav, A. K. K. Reddy, A. N. Reddy, and S. Ranjan, "Detecting algorithmically generated malicious domain names," in Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, 2010, pp. 48-61.

[39] D. Plohmann, K. Yakdan, M. Klatt, J. Bader, and E. Gerhards-Padilla, "A comprehensive measurement study of domain generating malware," in 25th USENIX Security Symposium (USENIX Security 16), 2016, pp. 263-278.

[40] S. Schüppen, D. Teubert, P. Herrmann, and U. Meyer, "{FANCI}: Feature-based automated {NXDomain} classification and intelligence," in 27th USENIX Security Symposium (USENIX Security 18), 2018, pp. 1165-1181.

[41] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Learning to detect malicious urls," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 2, no. 3, pp. 1-24, 2011.

[42] D. L. Marino and M. Manic, "Modeling and Planning Under Uncertainty Using Deep Neural Networks," IEEE Transactions on Industrial Informatics, vol. 15, no. 8, pp. 4442-4454, 2019, doi: 10.1109/TII.2019.2917520.

[43] X. Yan, Y. Xu, B. Cui, S. Zhang, T. Guo, and C. Li, "Learning URL Embedding for Malicious Website Detection," IEEE Transactions on Industrial Informatics, vol. 16, no. 10, pp. 6673-6681, 2020, doi: 10.1109/TII.2020.2977886.

[44] M. Alsaedi, F. A. Ghaleb, F. Saeed, J. Ahmad, and M. Alasli, "Cyber threat intelligence-based malicious url detection model using ensemble learning," Sensors, vol. 22, no. 9, p. 3373, 2022.

[45] J. Wang, S. Qian, J. Hu, and R. Hong, "Positive Unlabeled Fake News Detection Via Multi-Modal Masked Transformer Network," IEEE Transactions on Multimedia, pp. 1-11, 2023, doi: 10.1109/TMM.2023.3263552.

[46] Y. Chai, Y. Zhou, W. Li, and Y. Jiang, "An Explainable Multi-Modal Hierarchical Attention Model for Developing Phishing Threat Intelligence," IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 2, pp. 790-803, 2022, doi: 10.1109/TDSC.2021.3119323.

[47] S. Dadgar and M. Neshat, "A Novel Hybrid Multi-Modal Deep Learning for Detecting Hashtag Incongruity on Social Media," Sensors-Basel, vol. 22, no. 24, p. 9870, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/24/9870.

[48] Kaggel. "Malicious URLs dataset." Kaggel. https://www.kaggle.com/sid321axn/malicious-urls-dataset (accessed 01/05/2023, 2023).

[49] D. Ranganayakulu and C. C, "Detecting Malicious URLs in E-mail – An Implementation," AASRI Procedia, vol. 4, pp. 125-131, 2013/01/01/ 2013, doi: https://doi.org/10.1016/j.aasri.2013.10.020.

Mohammed AlSaedi was born in Saudi Arabia and received his BSEE and MSEE degrees in electrical engineering from King Saud University, Riyadh, Saudi Arabia, in 1995 and 2004, respectively. He completed his research for the PhD degree at the University of Dayton, Dayton, Ohio. His doctoral work has resulted in two journal papers and several conference presentations and articles in conference proceedings. Currently, he is working at Taibah university- Saudi Arabia. His research interests include network security, signal and image processing, and nonlinear optics.

Fuad A. Ghaleb obtained a B.Sc. degree in computer engineering from the Faculty of Engineering, Sana’a University, Yemen, in 2003, and M.Sc. and Ph.D. degrees in computer science (information security) from the Faculty of Engineering, School of Computing, Universiti Teknologi Malaysia (UTM), Johor, Malaysia, in 2014 and 2018, respectively. He is currently a Senior Lecturer with the Faculty of Engineering, School of Computing, UTM. His research interests include vehicular network security, cyber security, intrusion detection, data science, data mining, and artificial intelligence. He has been a recipient of many awards and recognitions, including the Postdoctoral Fellowship Award, the Best Postgraduate Student Award, the Excellence Awards, and the Best Presenter Award from the School of Computing, Faculty of Engineering, UTM, as well as the best paper awards from many international conferences

Faisal Saeed (Member, IEEE) is a Senior Lecturer in the Computing and Data Science Department at the School of Computing and Digital Technology, Birmingham City University (BCU), UK. He leads the smart health lab at Data Analytics and AI Research Group at BCU. Previously he worked as Assistant/Associate Professor at Taibah University, KSA from 2017-2021, and as Senior Lecturer at the Department of Information Systems, Faculty of Computing, Universiti Teknologi Malaysia (UTM), from 2014-2017. Faisal received his BSc in Computers (Information Technology) from Cairo University, Egypt, MSc in Information Technology Management, and Ph.D. in Computer Science from UTM, Malaysia. He has published several papers in indexed journals and international conferences. His research interests are data mining, artificial intelligence, machine learning, information retrieval, and health informatics.

Mohammed Alasali is an Assistant Professor in Computer Engineering at Taibah University, Saudi Arabia. He holds a Ph.D. in Computer Science and Engineering from King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia. His research interests include digital systems, IoT, AI, and data security. He has worked on more than four funded projects in the past three years at Taibah University. These projects cover a wide range of practical topics in the field of computer science and engineering, including machine learning, deep learning techniques in 2D and 3D images in dentistry, network security and application of Ai in public health.

Jawad Ahmad (Senior Member, IEEE) is an experienced Researcher with more than ten years of cutting-edge research and teaching experience in prestigious institutes, including Edinburgh Napier University, U.K.; Glasgow Caledonian University, U.K.; Hongik University, South Korea; and HITEC University Taxila, Pakistan. He has taught various courses both at Undergraduate (UG) and Postgraduate (PG) levels during his career. He has coauthored more than 100 research papers in international journals and peer-reviewed international conference proceedings. His research interests include cybersecurity, multimedia encryption, and machine learning. He regularly organizes timely special sessions and workshops for several flagship IEEE conferences. He is an invited reviewer for numerous world-leading high-impact journals (reviewed more than 100 journal articles to date).