Multimodal Classification of Onion Services for Proactive Cyber Threat Intelligence using Explainable Deep Learning

Harsha Moraliyage¹, (Member, IEEE), Vidura Sumanasena¹, Daswin De Silva¹, (Senior Member, IEEE), Rashmika Nawaratne¹, (Member, IEEE), Lina Sun¹, and Daminda Alahakoon¹, (Member, IEEE)

¹Centre for Data Analytics and Cognition, La Trobe University, Victoria, Australia.
Corresponding author: Harsha Moraliyage (e-mail: h.moraliyage@latrobe.edu.au)

ABSTRACT The dark web has been confronted with a significant increase in the number and variety of onion services of illegitimate and criminal intent. Anonymity, encryption, and the technical complexity of the Tor network are key challenges in detecting, disabling, and regulating such services. Instead of tracking an operational location, cyber threat intelligence can become more proactive by utilizing recent advances in Artificial Intelligence (AI) to detect and classify onion services based on the content, as well as provide an interpretation of the classification outcome. In this paper, we propose a novel multimodal classification approach based on explainable deep learning that classifies onion services based on the image and text content of each site. A Convolutional Neural Network with Gradient-weighted Class Activation Mapping (Grad-CAM) and a pre-trained word embedding with Bahdanau additive attention are the core capabilities of this approach that classify and contextualize the representative features of an onion service. We demonstrate the superior classification accuracy of this approach as well as the role of explainability in decision-making that collectively enables proactive cyber threat intelligence in the dark web.

INDEX TERMS Attention, Bahdanau, Cybersecurity, Cyber Threat Intelligence, Dark Web, Deep Learning, Grad-CAM, multimodal classification, Onion Services

I. INTRODUCTION
The Internet and associated cyberspace technologies have advanced human civilization into the information age. We are highly dependent on these technologies for our basic needs, services, transactions, social interactions, and lifestyle conveniences. Despite the numerous benefits of cyberspace, our dependence has become a vulnerability that can be readily exploited. Cyber Threat Intelligence (CTI) is a field of study that aims to identify and mitigate such malicious activities in cyberspace through data-driven processes and systematic analysis of voluminous files, binaries, events, and Open Source Intelligence (OSINT) to enhance cybersecurity. Recent developments in Artificial Intelligence (AI) have been effective in detecting most cyber threats with minimal human intervention which will improve cybersecurity while reducing the human errors [1]. However, cybersecurity threats originating and persisting in the dark web remain a persistent challenge due to the technical complexity while the interpretability of AI models used in dark web threat analysis applications hasn’t been explored well in the existing research literature [2][3]. The two main constituents of the Internet are the surface web and the deep web [4]. The deep web comprises of all the content which is not indexed by default web crawlers and secured by different types of authentication protocols. The dark web can be identified as the deepest layer of the deep web which is hidden from the default web search engines and operates using Tor’s HS protocol which hides the IP addresses of the hosting servers and clients who access the services [5]. These anonymous services which are exposed through the Tor communication network using a special top-level domain name called .onion are also known as onion services[5][6]. Anonymity of onion services makes it difficult for CTI application to trace client and server locations that increase the risk of cybersecurity threats coming from the dark web. Majority of onion services are operating in providing illicit and criminal services such as cyber hacking, drug trafficking, child pornography, and the sale of illegal firearms [7][8]. English is the primary language used in the onion sites while Russian, French, and Italian are the secondary languages [7]. Onion services on the dark web can be categorized into four major types namely hacker forums, carding shops, marketplaces, and internet relay chat (IRC) services [7].

Collaborative monitoring and sharing of information across cybersecurity agencies are currently the most effective
strategy to prevent or minimize cybersecurity threats[9]. Threat sharing platforms are critical in sharing information across multiple interested parties [10][11]. Large technology and cybersecurity organizations like IBM, CISCO, Crowdstrike, and McAfee, share and sell threat intelligence via commercial tools [10]. However, the dark web has experienced a multifold increase in illicit onion services over recent years [12][13]. Thereby, continuous monitoring of onion services for malicious content, leading to insights that are interpretable and explainable, remains a significant challenge.

In this paper, we propose a novel AI approach, multimodal classification using explainable deep learning to address this challenge. This will enhance the effectiveness of detecting cybersecurity threats originating from the onion services in the Tor communication network of the dark web. The proposed modular AI component which integrates the multimodal deep learning architecture can efficiently categorize the onion services in the Tor communication network to provide enhanced threat intelligence capabilities. Multimodal classification learns to classify inputs from multiple modalities, instead of a single modality. This has been effectively used in many domains, such as image captioning, sentence matching in images, and speech recognition [14][15][16][17]. The key contributions of the research presented in this paper are as follows.

- A novel multimodal deep learning approach for multi-label classification of diverse categories of onion services that integrates a Convolutional Neural Network (CNN) and word embedding with a transfer learning approach
- Extension of this approach for explainability and interpretability of classification outcomes using Gradient-weighted Class Activation Mapping (Grad-CAM) and Bahdanau additive attention mechanism
- Empirical evaluation of the proposed approach using a state-of-the-art onion services dataset [18] curated by the Computer Incident Response Center Luxembourg (CIRCL) containing more than 8000 instances of onion services across 51 categories
- Modularization of the proposed approach as an AI pipeline that can be plugged into threat analysis platforms like CIRCL’s Analysis Information Leak (AIL) [19] framework

The rest of the paper is organized as follows; Section 2 presents related work and the motivation behind this study. Section 3 delineates the proposed approach, a novel multimodal classification approach based on explainable deep learning that classifies onion services based on image and text content. This section also presents the modular architecture of the approach. The empirical evaluation, the dataset, results, and model explainability are reported in Section 4. A discussion of outcomes and concluding remarks are documented in Section 5.

II. RELATED WORK
The exponential growth of onion services poses a significant challenge for governments, organizations, and security agencies across the world, due to the anonymity and severity of cybersecurity threats originating from illegal activities on the dark web [12]. Analysis of dark web hacker forums that hackers used as an information exchange platform to share the strategies on hacking showed the importance of monitoring the dark web for proactive CTI to detect future possible cybersecurity threats [20][21]. Bitcoin and other cryptocurrencies fuel the dark web economy and act as the default payment option for illegal transactions across many onion services [5]. For instance, the sale of 500 million hotel records in a dark web forum for eight bitcoins is considered the most prolific information trafficking activity of all time in the information security space [22]. CTI is increasingly adopted by organizations and governments as it utilizes a data-driven approach to provide actionable insights on future cybersecurity threats using current threat intelligence [23]. The lack of well-defined interfaces and interoperable features between different types of threat sharing platforms is a key challenge in exchanging threat intelligence across multiple entities [24]. Recently developed threat sharing platforms have identified these gaps and have used open standards and protocols to allow wider community to adopt these tools on sharing cybersecurity threats [24]. CIRCL adopted open standards to develop a Malware Information Sharing Platform (MISP) [11] which is a threat sharing platform and AIL which is an information leak analysis framework to provide proactive CTI to improve the resistance against cybersecurity threats. A motivation for this research is the development of a built-in module that integrates with such a threat sharing platform.

![FIGURE 1. Standard dark web content classification pipeline](https://example.com/dark_web_pipeline.png)

Automatic classification of onion services is primarily
challenging due to the clandestine disposition of such services and their exponential growth[5][25]. Recent AI research for dark web onion services classification approaches has mainly used the textual content of such services while using both supervised learning methods and unsupervised learning methods [22]. The textual content of the onion service is represented with weighted feature matrices by using a bag of words (BoW) and term frequency-inverse document frequency (TF-IDF) models. This representation is then used to formulate a classification problem to categorize the onion services [22]. The lack of adequate training data to use in supervised learning methods has been identified as a challenge in the onion service classification methods [5][22]. Support Vector Machines (SVM), logistic regression, and Bayesian networks are identified as the most widely used classifiers in the onion service classification methods [22],[26].

Ghosh et al. [5] proposed an automated crawling system named “Automated Tool for Onion Labeling” (ATOL) to analyze and classify the content of public onion services. It consists of a novel keyword discovery algorithm (ATOLKeyword) to identify keywords that are difficult for humans to detect from text content [5]. Identified keywords are then used to build a classifier (ATOLClassify) by assigning a novel feature weighting algorithm to achieve a 12% accuracy improvement for F1 score [5]. They have also proposed a semi-supervised learning algorithm based on clustering (ATOLCluster) to group similar content. Xu et al. [27] proposed an ontology model that uses weighted feature category ontology to classify dark web data sources and achieved a precision value of 91.6% and a recall value of 92.4%. A study carried out by Graczyk et al. [28] used the information of products sold in the popular anonymous online market Agora on the dark web to build a product classifier using TF-IDF feature extraction and SVM that achieved 79% accuracy. Sabbah et al. [29] implemented a hybrid term weighting technique that uses an ensemble strategy to calculate weights using multiple weighting matrices including TF-IDF, entropy, and Glasgow to achieve better accuracy compared to individual term weighting technique on classifying dark web forums. Dalins et al. [30] introduced Tor-use Motivation Model (TMM) which defined two dimensional labeling scheme to categorize onion services. Al Nabki et al. [31] presented a dataset called DUTA which contains of 6831 onion service contents that have been manually labeled into 26 categories and proposed a classification pipeline based on the text content of the onion services. They have used TF-IDF and BoW together with three widely used supervised learning classifiers namely SVM, logistic regression, and Naive Bayes in the proposed classification pipeline. Noor et al. [12] proposed a technique called “Query Probing” to automatically classify content extracted from deep web sources. He et al. [32] proposed a schema-based clustering approach with novel clustering objecting function named “model-differentiation” to classify deep web sources. Buldin et al.[33] implemented k-nearest neighbors algorithm (kNN) based text categorization method to classify Russian language onion-sites. Burda et al. [3] developed a tool named MASSDEAL to automatically explore and categorize the onion services based on their textual content.

Most of the existing research studies on onion service classification were mainly focused on the textual content while providing less prominence for the visual features of the onion service. The pipeline approach shown in Fig.1 is widely adopted in building a dark web classification model. Content of the onion services can be arranged in different variations of user interface (UI) designs. Some onion sites may only contain visual images without any textual content which can more frequently be observed among pornography, cryptocurrency, and payment panel-oriented onion services [34]. Hence, a classification strategy which solely based on the textual content wouldn’t work for every type of onion service. Bag of Visual Words (BoVW) and perceptual hashing techniques are popular among existing classification models which use the visual contents of the onion service. Fidalgo et al. [35] proposed a BoVW model to classify frequently found illegal images in onion services which were experimented on TOIC (TOR Image Categories) dataset. Biswas et al. [34] presented two classification pipelines based on perceptual hashing with hamming distance and BoVW with SVM to classify snapshot images of six active TOR domains found in the public dataset of Darknet Usage Service Images (DUSI).}

![Image 37x751 to 114x765](image1.png)

**FIGURE 2.** An example of an onion service dealing in counterfeit currency.

Existing research literature has not explored the capabilities
of deep learning for onion service classification. Current research has mostly concentrated on improving the accuracy of existing models, with new techniques receiving little attention. Innovative studies on transfer learning have helped deep neural networks to work with fewer data and achieve better accuracy. The Explainability of predicted outcomes in onion service classification is also underexplored. Our study focuses on classification and the explainability capabilities of multimodal deep learning. Classification of onion sites based on publicly available taxonomy enables better shareability across different threat sharing platforms which provide the collaborative power to combat such cybersecurity threats. Hence, we also focus our attention on the usability of our research in adopting real-world platforms by proposing a modular architecture that can be plugged into threat sharing platforms.

III. PROPOSED APPROACH

The image and text modalities contain the main features required for the detection and classification of an onion service. As demonstrated in Fig.2, the images represent counterfeit notes, passports, credit cards, while the text phrases such as counterfeitcentre, bitcoins, and passport are further indicative of the classification task. The proposed multimodal classification approach based on explainable deep learning consists of two pathways for the two modalities of images and text. It is illustrated in Fig.3.

Each pathway has a learning phase followed by an explainability phase and then merged for the final phase of multi-label classification. Unavailability of enough training data is a common problem in certain application domains including onion service classification [22]. The need for a large amount of training data is one of the major drawbacks in deep learning applications [36]. But the transfer learning has emerged as a solution to this problem which uses a knowledge transferring approach where it uses knowledge gathered from one domain to solve a problem in another domain [36]. Each pathway utilizes pre-trained models to get the benefit of transfer learning and is discussed in detail in the following subsections.

A. PATHWAY 1 – LEARNING THE IMAGE MODALITY

Pathway 1 of the proposed model in Fig.3 uses the visual content of the onion screen capture to predict the class labels. CNNs deep learning architectures are effectively used to address wide variety of computer vision problems [37]. Natural visual perception of the human eye and the functionality of the visual cortex is the main inspiration behind the origination of CNNs [38]. A generic CNN contains three major types of layers which are named as convolution, pooling, and fully connected layers.

We have applied transfer learning in Pathway 1 by using a pre-trained CNN network to improve the performance and the accuracy of the model with a low volume of training data. Most well-known deep CNN architectures including ResNet, VGGNet, and Inception networks contain millions of trainable parameters which require a massive amount of training data.
data, computational resources, and time to build a classifier that matches the state of the art (SOTA) performance [39]. ImageNet is a popular labeled dataset that contains more than 1.2 million images belonging to over 1000+ categories [39]. Researchers and organizations have used ImageNet data to build classifiers based on popular deep CNN architectures where they have saved the weights of the trained deep CNN and made it available for the community to support transfer learning applications. We have experimented our proposed model with three pre-trained CNN architectures namely VGG16, ResNet50, and InceptionV3 each of which is trained on ImageNet dataset. As in Fig.3, pre-trained CNN model in the image modality is replaced with each of these models to extract features and evaluate the final performances of the proposed multimodal architecture.

During the model training, the last few layers of the pre-trained CNN network are set to trainable to fine-tune the existing weights. The early convolution layers of the CNN network capture the generic features of an input which is common for any type of image content. Hence, pre-trained weights of early layers of the CNN network are used without any adjustment which also improves the model training time. As in Fig.3, onion screen captures represented through a multi-dimensional array of pixel values with three color channels are fed into a pre-trained CNN network which outputs a set of feature maps at each convolutional layer. These feature maps are generated by kernels known as filters which scan through the image to capture unique features of the onion screen capture which influence the target predictions. Early convolution layers of the pre-trained network detect the edges and curves of the input image while later convolution layers capture more abstract features such as high-level images present in the onion screen capture [38]. A pooling layer is followed by one or more convolution layers used to reduce the dimensionality of feature maps. Max pooling is the widely adopted strategy in pooling layers which shrinks the feature maps by emitting the highest feature map values among the defined neighborhood [40]. These layers will highlight the most influential areas of the onion screen capture towards the predictions. The last layers of the CNN network use a set of fully connected layers as in generic DNN to output the predictions. Convolution layers and fully connected layers use non-linear activation functions to add non-linearity on the captured features of the onion screen capture. Intermediate layers of the proposed model use Rectified Linear Unit (ReLU) function defined in (1) as the activation function which is a widely used non-linear activation function due to the lower computation cost [39].

\[
f(x) = \max(0, x)
\]

Many deep learning models are motivated by neuroscientific advancements which try to mimic the functionality of the human brain [40]. Yet, the explainability of deep learning models is far from achieving human-level performance [41]. The ability to justify the predicted outcomes gives confidence for decision makers to trust the output of predictive models which can be crucial in mission critical applications [41]. Identifying the regions of the image that put more weight towards target predictions used as the main technique to provide explainability in CNN networks [41]. Zhou et al. [42] proposed a technique called class activation maps (CAM) to identify regions used by the deep learning model to produce its output. But CAM isn’t capable of handling deep learning networks that contain fully connected layers. Grad-CAM is a more generalized form of CAM which can be applied to any neural network which contains a convolutional structure [43], [44]. We have used Grad-CAM technique in Pathway 1 to explain the outcomes of the predictions by highlighting the regions via heatmaps to provide visual explanations. Most commonly the last convolution layer is used to compute and visualize the Grad-CAM output. We have added a new convolution layer ConvD (Fig.3) to visualize the Grad-CAM output. During the Grad-CAM calculation, it calculates gradients of predicted onion class label \( c \) which is having the largest predicted probability with respective to feature maps \( A \) of the selected convolutional layer ConvD. These gradients are global average pool over width \((W)\) and height \((H)\) dimensions with normalizing factor \(( Z)\) to obtain neuron importance weights as in (2) [43]. \( k \) denotes the number of output feature maps of selected convolution layer ConvD. Calculated weights represent partial linearization from downstream stream feature maps of selected convolution layer and capture the importance of feature map \( k \) towards target prediction onion class label \( c \) [43]. ReLU activation function applied to obtain positively influence features towards the prediction to calculated Grad-CAM over calculated weights as in (3). The final output used to visualize and identify the regions with more importance which we have leveraged in our study to provide explanations.

\[
w_k^c = \frac{1}{Z} \sum_{i=1}^{W} \sum_{j=1}^{H} \frac{\partial y^c}{\partial A_{ij}^k}
\]

\[
I_{Grad-CAM} = ReLU \left( \sum_{k=1}^{K} w_k^c \cdot A^k \right)
\]

B. PATHWAY 2 - LEARNING THE TEXT MODALITY

In pathway 2, the text content of the onion screen capture is processed through word embedding and attention layers to provide target predictions. Word embedding is at the heart of solving many Natural Language Processing (NLP) problems including sentiment analysis, text summarization, and question answering [45],[46]. Word embeddings can be described as a vector space model which represents words with fixed-length vectors based on the co-occurrence statistics that capture the characteristics of words based on their distributions in the text content [47]. Words with similar meanings tend to appear more frequently in similar contexts which is the main...
intuition behind the development of word embeddings [46]. Mikolov et al. [48] proposed the Continuous Bag of Words (CBOW) and skip-gram model to train word embeddings that lead to the creation of strong word embedding representations. Existing studies discussed in the related work section highlighted the importance of using the text content of the onion service to assist with predicting the target class labels. Traditional models such as SVMs and logistic regression were used with BoW and TF-IDF representations to solve many NLP problems which were very common in onion service classification applications [45]. But word embeddings capture both syntactical and semantic representation of a text which BoW models aren’t capable of handling within traditional models [46]. Even shallow DNNs outperform traditional models in solving NLP problems [46]. Hence, we have used a word embedding layer in the proposed model to represent the input text content of onion capture to obtain better text representation. Word embedding layer is used as the first layer of most of the DNN architectures which transform input words to their vector space representations as shown in Fig.4.

![FIGURE 4. Transformation of input word sequence through the word embedding layer.](image)

We have used a pre-trained word embedding layer in pathway 2 without training word embeddings from scratch and fine-tune the pre-trained word embeddings layer to gain the benefit of transfer learning. Pre-trained word embeddings which have been trained based on complex architectures and large text corpus could provide better accuracy on NLP tasks and requires a lower volume of training data. Pre-trained word embeddings can even capture the similarities of the words that do not appear in the training dataset [49]. GloVe, BERT, ELMo, and OpenAI GPT are well-known word embedding architectures that capture complex characteristics and representations of the words in the text [50]. We have trained and compare the performance of the proposed model with Word2Vec [48], GloVe [50], and fastText [51] pre-trained word embeddings model weights. The vocabulary used in the NLP task act as a dictionary that contains all the unique words of the training dataset during the training process. We have used a pre-defined vocabulary size with the most frequent words of the text corpus of onion service captures without using the entire set of unique words. The input length of the text determines how many words will be input to the network. We have used pre-defined input length with post padding and truncation. Post padding use to fill the text content in case of it being less than the specified input length while post truncate removes the words which comes after the input length. Words which are not present in the vocabulary have been replaced with <OOV> token as in Fig.4. Words which are not present in pre-trained word embeddings weights replaced with random word vectors which will be fine-tuned during the training process.

Humans pay attention to certain parts of a picture or the keywords of the text to capture or recognize the content [52]. Attention in DNNs is inspired by human attention, which focuses on the most relevant parts of the input which is then used to progress into the next layers [52]. This functionality improves the explainability of a neural network by identifying the features that receive more attention towards the target output predictions [52]. Therefore, we have utilized the attention mechanism in our proposed multimodal neural network to identify the areas of the text content which receive more attention to explain the predicted outputs. The attention mechanism used in our model is inspired by the self-attention model proposed by Lin et al. which was used to understand the interpretability of sentence embeddings [47]. We have used Bahdanau [53] attention which is an additive attention mechanism compared to the self-attention and is widely used in Neural Machine Translation (NMT) network architectures [52]. The attention mechanism used in our multimodal network architecture can be visualized as in Fig.5 which comes after the word embedding layer.

![FIGURE 5. Bahdanau additive attention mechanism used in the proposed approach.](image)
The attention mechanism in the proposed model has the objective of capturing the context of the words in onion service content that appear together to identify the unique services offered through the site. Attention scores respective to the input features indicate the feature importance hence visualizing attention scores provides a way to interpret the results and improve the explainability of the model. The proposed attention model as in Fig.5 has two main parts. The first part of the attention model uses a Bidirectional Long short-term Memory (Bi-LSTM) layer and then uses an attention layer to calculate the context vector representation of the input word embeddings. LSTM layers are generally used to capture contextual relationships on a long sequence of data and are capable of mitigating the vanishing gradient problem of Recurrent Neural Networks (RNNs) [54]. The use of Bi-LSTM in the proposed model has the objective of capturing the interdependencies between words represented through the word embeddings [52].

Suppose our extracted text of the onion service screen capture has a length of \( n \) and has a word embedding dimension length of \( d \). Then input text can be represented as in (4) which is a \( n \) by \( d \) dimensional vector.

\[
S = (w_1, w_2, w_3, ..., w_n)
\]

Bidirectional LSTM layer generates hidden states representing contextual relationships between words on its forward and backward scan through input sequence \( S \) as in (5) and (6) where \( w_t \) represent the input word embeddings of \( t \)th location word and \( h_{t-1} \) represent the hidden state of LSTM in the previous step of forwarding pass and \( h_{t+1} \) indicate the hidden state of the backward pass of the LSTM layer. A combination of these values is used to calculate the complete hidden state of \( h_t \).

\[
\vec{h}_t = \overline{\text{LSTM}}(w_t, \vec{h}_{t-1})
\]

\[
\vec{h}_t = \overline{\text{LSTM}}(w_t, \vec{h}_{t+1})
\]

The final hidden state is obtained by combining \( \vec{h}_t \) and \( \vec{h}_t \) indicate the complete hidden state at \( t \) time step. If a single LSTM unit contains \( u \) number of hidden units, then the final output of complete hidden bidirectional LSTM layer \( H \) can be presented as in (7) which will have the dimension of \( n \) by 2u.

\[
H = (h_1, h_2, h_3, ..., h_n)
\]

The attention vector \( A \) shown in Fig.5 contains the attention scores for the input text embeddings which is calculated using (8). It uses the transpose of complete hidden state \( H \) and LSTM layer output \( S \) and a set of trainable weight vectors \( W_{s1}, W_{s2}, W_{s3} \).

\[
A = \text{softmax}(W_{s3} \tanh(W_{s1}H^T + W_{s2}S))
\]

The calculated attention scores are a representation of the attention towards the target prediction. The length of the attention vector is equal to the max length \( n \) of the input text. We have used this approach in our study to explain the outputs of the multimodal network predictions with respect to the text input of onion site content. These attention scores are then used to build the context vector shown in Fig.5 which feeds into the next layer of the neural network. The context vector \( C \) in (9) is calculated using the weighted sum of all hidden states \( H \) and their corresponding attention scores \( A \). This mechanism helps the context vector to capture the information from the entire input sequence and focus on the relevant positions in the input text sequence [55]. Due to the better alignment in calculating the context vector, it will improve both the performance and quality of the final output[55]. The length of the context vector is equal to the dimension of the hidden state of LSTM cell.

\[
C = \sum A.H
\]

The last fully connected layers of pathway 1 and pathway 2 are combined via a concatenation layer to provide target predictions. Each pathway captures important features relevant to its input data which influence the target predictions.

**Algorithm 1: Multimodal Classification Approach**

1. **Input:** \( X_{\text{train}}, X_{\text{test}} \) ← Corpus of train and test images  
   \( L_{\text{train}}, L_{\text{test}} \) ← Corpus of train and test data labels  
   \( H \) ← Values of hyperparameters
2. **Output:** \( M \) ← Trained multimodal classifier  
   \( A_{\text{test}} \) ← Accuracy based on F2-Score of test data  
   \( T_{\text{test}} \) ← Attention scores of test data  
   \( G_{\text{test}} \) ← Grad-CAM heat map values of test data  
   \( \hat{L}_{\text{test}} \) ← Predicted class labels of test data
3. **Train**  
   for each \( d \) in \( X_{\text{train}} \) do
4. \( n_{\text{train}} := \) extract text content using OCR for \( d \)
5. \( n'_{\text{train}} := \) translate text content using NMT for \( n_{\text{train}} \)
6. \( N_{\text{train}} := \) pre-process text content for \( n'_{\text{train}} \)
7. end
8. \( M := \) initialize multimodal classifier with \( H \)
9. train (\( M, X_{\text{train}}, N_{\text{train}} \))
10. **Evaluate and explainability**
   for each \( d \) in \( X_{\text{test}} \) do
11. \( n_{\text{test}} := \) extract text content using OCR for \( d \)
12. \( n'_{\text{test}} := \) translate text content using NMT for \( n_{\text{test}} \)
13. \( n''_{\text{test}} := \) pre-process text content for \( n'_{\text{test}} \)
14. \( \hat{L}_{\text{test}} := \) predict (\( M, d, n''_{\text{test}} \))
15. \( T_{\text{test}} := \) attention_scores (\( M, d, n''_{\text{test}} \))
16. \( G_{\text{test}} := \) grad_cam_heat_maps (\( M, d, n''_{\text{test}} \))
17. \( A_{\text{test}} := \) F2_score (\( \hat{L}_{\text{test}}, L_{\text{test}} \))
18. return \( M, \hat{L}_{\text{test}}, A_{\text{test}}, T_{\text{test}}, G_{\text{test}} \)
C. MODELING APPROACH TO FUSE MODALITIES

The output of a multimodal classifier uses the features extracted from each modality to perform the classification task. Hence classifier accuracy will be improved as it’s receiving information from multiple modalities compared to using a single modality input[56]. We are using the features extracted from text and image modality to classify the onion service into multiple categories. The goal of the multimodal classifier is to minimize the cost function $J$ in (11) of model outputs to its ground truth labels of onion sites. The loss $L$ of a single training instance is calculated as in (10) where $C$ is the number of classes, $y_j$ is the ground truth label and $\hat{y}_j$ is the predicted label. The cost function $J$ which is average over the entire training set of size $N$ is calculated using (11).

$$L(\hat{y}, y) = -\sum_{j=1}^{C} y_j \log (\hat{y}_j) + (1 - y_j) \log (1 - \hat{y}_j)$$

$$J = \frac{1}{N} \sum_{i=0}^{N} L(\hat{y}^{(i)}, y^{(i)})$$

A classifier which extracts information using pre-trained embeddings (Glove, FastText, Word2Vec) from textual inputs of an onion site contents can be represented as in (12) where $W$ and $T$ are weight matrices and $x_n^t$ is the matrix containing textual features representation. But this ignores the features of the visual content of the onion site.

$$a(x_n^t) = WTX_n^t$$

A classifier which extracts the visual features using pre-trained CNN network (VGG, ResNet, Inception) can be represented as in (13) where $W$ and $U$ are weight matrices and $x_n^v$ in (13). This ignores the informing receiving from the textual input of the onion site.

$$a(x_n^v) = WUX_n^v$$

In our multimodal classifier, we have used an additive approach by combining the features received from each modality as in (14).

$$a(x_n) = W(Tx_n^t + Ux_n^v)$$

Algorithm 1 outlines the novel multimodal classification approach with explainable outcomes used in this paper.

D. MODULARIZATION OF THE PROPOSED APPROACH FOR THREAT ANALYSIS PLATFORMS

The proposed approach is modularized as an AI pipeline to be plugged into threat analysis platforms. The proposed modular architecture is presented in Fig.6. It begins with a dark web data extractor which captures all the content in onion services and then an Optical Character Recognition (OCR) module separates written text from images. In the written text, all non-English content is translated using the Neural Machine Translation (NMT) module. The NMT module can be implemented using widely cited cloud services such as those provided by Google or Amazon. Next, the images and text are further separated for preprocessing tasks such as image resizing, lemmatization, and text tokenization, and then fed into the learning pathways depicted Fig.6, pathway 1 for image modality learning and pathway 2 for text modality. Following this, the explainability phase transforms the learning output into interpretable outcomes that are sent across to the multi-classification module. This is the final module which generates the predictions and explanations of the classification outputs for the threat analysis platform.

IV. EXPERIMENTS

We conducted an empirical evaluation of the proposed approach on the state-of-the-art onion services dataset [18] curated by the Computer Incident Response Center Luxembourg (CIRCL) containing more than 8000 instances of onion services across 51 categories. Given the imbalanced distribution of taxonomy labels as in Fig.7, we selected instances that have label frequency greater than 150 with the inclusion of special categories as the final dataset, which contains 5805 captures and expands across 27 taxonomy labels.
As an example, onion screen capture in Fig.2 is labeled with taxonomies of dark-web:topic="counteir-feit-materials", dark-web:topic="finance-crypto" and dark-web:motivation="marketplace-for-sale". Reasons behind the labeling can be identified as this onion site offers services by selling fake money notes which are counterfeit-materials through a marketplace, and it accepts bitcoin payments. Some minor inconsistencies found in the tagged labels of the dataset have been fixed by correcting them through a manual inspection. We have also added finance-crypto taxonomy label to relevant onion screen captures as this term was introduced recently and wasn’t present in the original dataset. This label is used to associate onion sites that use cryptocurrency services and payments. Duplicate screen captures of some sites were filtered out and removed during the data pre-processing phase.

We have used Tensorflow 2.1.0 and Python 3.7.0 to implement the proposed explainable multimodal deep learning architecture. VGG16, ResNet50 and InceptionV3 CNNs implementations in the Tensorflow library with pre-trained weights of the ImageNet dataset have used to extract the features of the onion site image content as specified in Fig.3.

We have used pre-trained word vectors with 300 dimensions from Glove, FastText, and Word2Vec word embeddings methods to compare the performance of our proposed model. The hidden state size of the Bi-directional LSTM cell is set to half of the embedding dimension which provides an adequate space to capture the capture contextual relationships of word embedding vectors. F2 score calculated as in (18) is used as the model performance evaluation measure in our experiment. It favors recall over the precision value [58]. Precision indicates the model performance in predicting positive class while recall indicates the model performance in predicting positive class from actual positive classes. In a multilabel classification problem, binary classification accuracy does not provide insights of model performance. F1 score captures the average of precision and recall. F2 is a more generalized form of F1 score which uses value to control the importance of recall over precision and uses default value of

| Loss Function | Binary cross-entropy |
|---------------|----------------------|
| Vocabulary sizes | 10000, 15000, 20000 |
| Max text lengths | 150, 200, 250, 300 |
| Input image | (150,150,3) |
| Activation functions | ReLU, Sigmoid (Last layer) |
| Evaluation Matrix | F2 Score |
| Optimizer | Adam |
| Learning Rates | 0.01, 0.001, 0.0001 |
| Epochs | 20 |
| Dropout Rate | 0.3 |

We have used Tensforflow 2.1.0 and Python 3.7.0 to implement the proposed explainable multimodal deep learning architecture. VGG16, ResNet50 and InceptionV3 CNNs implementations in the Tensorflow library with pre-trained weights of the ImageNet dataset have used to extract the features of the onion site image content as specified in Fig.3.

We have used pre-trained word vectors with 300 dimensions from Glove, FastText, and Word2Vec word embeddings methods to compare the performance of our proposed model. The hidden state size of the Bi-directional LSTM cell is set to half of the embedding dimension which provides an adequate space to capture the capture contextual relationships of word embedding vectors. F2 score calculated as in (18) is used as the model performance evaluation measure in our experiment. It favors recall over the precision value [58]. Precision indicates the model performance in predicting positive class while recall indicates the model performance in predicting positive class from actual positive classes. In a multilabel classification problem, binary classification accuracy does not provide insights of model performance. F1 score captures the average of precision and recall. F2 is a more generalized form of F1 score which uses value to control the importance of recall over precision and uses default value of

| Loss Function | Binary cross-entropy |
|---------------|----------------------|
| Vocabulary sizes | 10000, 15000, 20000 |
| Max text lengths | 150, 200, 250, 300 |
| Input image | (150,150,3) |
| Activation functions | ReLU, Sigmoid (Last layer) |
| Evaluation Matrix | F2 Score |
| Optimizer | Adam |
| Learning Rates | 0.01, 0.001, 0.0001 |
| Epochs | 20 |
| Dropout Rate | 0.3 |

We have used Tensforflow 2.1.0 and Python 3.7.0 to implement the proposed explainable multimodal deep learning architecture. VGG16, ResNet50 and InceptionV3 CNNs implementations in the Tensorflow library with pre-trained weights of the ImageNet dataset have used to extract the features of the onion site image content as specified in Fig.3.

We have used pre-trained word vectors with 300 dimensions from Glove, FastText, and Word2Vec word embeddings methods to compare the performance of our proposed model. The hidden state size of the Bi-directional LSTM cell is set to half of the embedding dimension which provides an adequate space to capture the capture contextual relationships of word embedding vectors. F2 score calculated as in (18) is used as the model performance evaluation measure in our experiment. It favors recall over the precision value [58]. Precision indicates the model performance in predicting positive class while recall indicates the model performance in predicting positive class from actual positive classes. In a multilabel classification problem, binary classification accuracy does not provide insights of model performance. F1 score captures the average of precision and recall. F2 is a more generalized form of F1 score which uses value to control the importance of recall over precision and uses default value of

| Loss Function | Binary cross-entropy |
|---------------|----------------------|
| Vocabulary sizes | 10000, 15000, 20000 |
| Max text lengths | 150, 200, 250, 300 |
| Input image | (150,150,3) |
| Activation functions | ReLU, Sigmoid (Last layer) |
| Evaluation Matrix | F2 Score |
| Optimizer | Adam |
| Learning Rates | 0.01, 0.001, 0.0001 |
| Epochs | 20 |
| Dropout Rate | 0.3 |

We have used Tensforflow 2.1.0 and Python 3.7.0 to implement the proposed explainable multimodal deep learning architecture. VGG16, ResNet50 and InceptionV3 CNNs implementations in the Tensorflow library with pre-trained weights of the ImageNet dataset have used to extract the features of the onion site image content as specified in Fig.3.

We have used pre-trained word vectors with 300 dimensions from Glove, FastText, and Word2Vec word embeddings methods to compare the performance of our proposed model. The hidden state size of the Bi-directional LSTM cell is set to half of the embedding dimension which provides an adequate space to capture the capture contextual relationships of word embedding vectors. F2 score calculated as in (18) is used as the model performance evaluation measure in our experiment. It favors recall over the precision value [58]. Precision indicates the model performance in predicting positive class while recall indicates the model performance in predicting positive class from actual positive classes. In a multilabel classification problem, binary classification accuracy does not provide insights of model performance. F1 score captures the average of precision and recall. F2 is a more generalized form of F1 score which uses value to control the importance of recall over precision and uses default value of

| Loss Function | Binary cross-entropy |
|---------------|----------------------|
| Vocabulary sizes | 10000, 15000, 20000 |
| Max text lengths | 150, 200, 250, 300 |
| Input image | (150,150,3) |
| Activation functions | ReLU, Sigmoid (Last layer) |
| Evaluation Matrix | F2 Score |
| Optimizer | Adam |
| Learning Rates | 0.01, 0.001, 0.0001 |
| Epochs | 20 |
| Dropout Rate | 0.3 |

We have used Tensforflow 2.1.0 and Python 3.7.0 to implement the proposed explainable multimodal deep learning architecture. VGG16, ResNet50 and InceptionV3 CNNs implementations in the Tensorflow library with pre-trained weights of the ImageNet dataset have used to extract the features of the onion site image content as specified in Fig.3.

We have used pre-trained word vectors with 300 dimensions from Glove, FastText, and Word2Vec word embeddings methods to compare the performance of our proposed model. The hidden state size of the Bi-directional LSTM cell is set to half of the embedding dimension which provides an adequate space to capture the capture contextual relationships of word embedding vectors. F2 score calculated as in (18) is used as the model performance evaluation measure in our experiment. It favors recall over the precision value [58]. Precision indicates the model performance in predicting positive class while recall indicates the model performance in predicting positive class from actual positive classes. In a multilabel classification problem, binary classification accuracy does not provide insights of model performance. F1 score captures the average of precision and recall. F2 is a more generalized form of F1 score which uses value to control the importance of recall over precision and uses default value of

| Loss Function | Binary cross-entropy |
|---------------|----------------------|
| Vocabulary sizes | 10000, 15000, 20000 |
| Max text lengths | 150, 200, 250, 300 |
| Input image | (150,150,3) |
| Activation functions | ReLU, Sigmoid (Last layer) |
| Evaluation Matrix | F2 Score |
| Optimizer | Adam |
| Learning Rates | 0.01, 0.001, 0.0001 |
| Epochs | 20 |
| Dropout Rate | 0.3 |
two in model performance evaluation [58]. In our imbalance dataset, we prefer recall over precision as outputs of false positive classes could provide decision makers to analyze and conclude on the decision.

The hyperparameters are tabulated in Table 2. Since this is a multi-label classification problem, we have used sigmoid activation function instead of softmax with binary cross-entropy loss at the last layer. We have preprocessed the images to match with the input size of pre-trained image models. Data augmentation is used in the training dataset which consists of onion screen captures to increase the size of the training dataset together with dropout layers to reduce the overfitting. We have experimented our approach for each of the three types of inputs: image only, text only, and multimodal (image and text). We have evaluated the performance of the proposed model with different vocabulary lengths, learning rates, and dropout rates. We have selected the Adam which is one of the widely adopted optimizers in our multimodal classifier due to its improved capabilities compared to existing optimizers including faster training times [59][60]. We have tested Adam optimizer with three learning rate values: 0.01, 0.001, 0.0001. The learning rate value of 0.01 gave better performance compared to the two other learning rates. We have observed a similar level of accuracy measures when we experiment with different levels of vocabulary lengths as in Table 2. Hence, we used a vocabulary length of 15000 for our experiments. Varying the max input length from 150 to 200 increased the validation dataset accuracy. But validation accuracy remained at similar levels when we vary the input lengths from 200 to 300 with 50 words increments. Hence, we selected the maximum input length as 250. Each model trained up to a maximum of 20 epochs. Training a model longer can decrease the model generalization and started to overfit to the training data. Hence, we have used early stopping callback of Tensorflow framework to halt training. Early stopping callback monitors the validation error after each epoch. It will stop the training process when validation error started to increase.

\[
\text{PRECISION} = \frac{\text{TRUE POSITIVE}}{\text{TRUE POSITIVE} + \text{FALSE POSITIVE}} \tag{15}
\]

\[
\text{RECALL} = \frac{\text{TRUE POSITIVE}}{\text{TRUE POSITIVE} + \text{FALSE NEGATIVE}} \tag{16}
\]
\[
F_1 = \frac{2 \times (\text{PRECISION} \times \text{RECALL})}{\text{PRECISION} + \text{RECALL}} \quad (17)
\]

\[
F_2 = (1 + \beta^2) \frac{(\text{PRECISION} \times \text{RECALL})}{\beta^2 \text{PRECISION} + \text{RECALL}} \quad \beta = 2 \quad (18)
\]

V. RESULTS

Results obtained in Table 3 are based on a learning rate value of 0.001 and vocabulary size of 15000 with a 0.3 dropout rate. These values are decided based on the experiments with varying parameters. The proposed multimodal architecture has better testing and validation accuracies compared to text and image only input models. The highest F2 accuracy of 0.935 for the training dataset and 0.785 for the validation achieved using Glove pre-trained word embeddings and VGG16 pre-trained model. Considering the validation accuracies of single modality inputs, VGG16 achieved better performance compared to ResNet50 and InceptionV3 models in image only inputs. GloVe word embeddings performed best among FastText and Word2Vec in text only input models. The text-only pre-trained word embeddings models performed considerably better compared to image only input models. VGG16 performed better in capturing the features of the input screen captures compared to ResNet50 while InceptionV3 model performed poorly.

We have identified a set of misclassified screen captures to diagnose issues of the proposed model and the dataset. During the analysis of the network predictions, we have observed some false positive predictions for the screen captures which is expected as F2 favors the recall. Most of the screen captures in ground truth labels only belong to an average of 2 to 3 labels. Some of the false positive predictions provide interesting explanations which we have identified through the Grad-CAM and attention visualization. The results shown in Fig. 8 are based on the highest accuracy multimodal network which trained in VGG16 and Glove word embeddings. The heatmaps shown in Fig. 8 indicate the explainability scores of the model inputs respective to the text and image modalities towards the target predictions. The impact of the features in each input increases when the color gradient change from black to light in the text modality and green to yellow in the image modality. Model output the predictions for example (a) in Fig.8 as drugs-narcotics and marketplace-for-sale where ground value only contains drugs-narcotics label. The Grad-CAM heatmap put more focus on the top left region which is indicated by the high feature map value on the yellow color region of the heatmap of the capture. The top left part contains cannabis leaf while attention scores of terms weed, cocaine, and mdma have a high score which further strengthens towards drugs prediction. Higher attention scores on checkout, product, cart, dashboard, and item indicate a key feature of a marketplace that provides explanations towards the model prediction of marketplace-for-sale.

The third example (c) in Fig.8 contains an onion site that sells fake credit cards and contains label values of credit-card, finance, and marketplace-for-sale. But our model predicted four labels that can be identified as credit-card, finance, marketplace-for-sale, and finance-crypto. When we analyze the attention scores keyword terms bitcoin and btcusd (bitcoin) have high attention scores as in Fig.8 Therefore, the model output which predicts finance-crypto can be rationalized with more meaningful explanations. The last example (d) shown in Fig.8 is related to a site that sells weapons. The ground truth labels of this site include LoginForms, captcha, and weapons and the model predicted all the ground truth labels. Grad-CAM output explains model output towards LoginForms as more heat is put on the user login area. Attention scores of terms gun, ammo, login, username, and slightly higher value on captcha provide a reasonable explanation for words with more attention towards the target prediction. We have found few examples which predict finance and credit-cards labels interchangeably. We have checked model explanations in this situation where the model predicts the output finance instead of ground truth label credit-card. During analysis of model explainability, we have noticed that terms cards, paypals, counterfeiterated has more attention scores.

The onion site which is related to gambling shown in example (b) of Fig.8 has the labels of gambling and finance-crypto which were predicted accurately by our model. The Grad-CAM output provides more importance towards the middle part of the image. Attention scores provide very strong explanations towards model explainability by having higher attention scores on terms bet, usdt (abbreviation of tether cryptocurrency), tether; and reasonable score for casino.

| Table 3: TRAINING AND VALIDATION ACCURACIES |
|-----------------------------------------------|
| Type     | Text  | Image | Train | Val     |
|----------|-------|-------|-------|---------|
| Text     | Glove | -     | 0.927 | 0.778   |
|          | Word2Vec | -   | 0.925 | 0.776   |
|          | FastText |     | 0.910 | 0.748   |
| Image    | -     | VGG16 | 0.849 | 0.611   |
|          | -     | ResNet50 | 0.848 | 0.587   |
|          | -     | InceptionV3 | 0.827 | 0.184   |
| Image+Text | Glove | VGG16 | 0.935 | 0.785   |
|          | Glove | ResNet50 | 0.920 | 0.782   |
|          | Glove | InceptionV3 | 0.926 | 0.741   |
|          | Word2Vec | VGG16 | 0.928 | 0.774   |
|          | Word2Vec | ResNet50 | 0.931 | 0.776   |
|          | Word2Vec | InceptionV3 | 0.933 | 0.715   |
|          | FastText | VGG16 | 0.916 | 0.762   |
|          | FastText | ResNet50 | 0.929 | 0.756   |
|          | FastText | InceptionV3 | 0.893 | 0.672   |
Label *finance* is a more general term compared to *credit-card* which is associated with any type of financial activity including selling counterfeit credit cards. Hence predicting the *finance* label instead of *credit-card* was revealed through the attention visualization. This is a practical example of analyzing the attention values towards target predictions. We showed that model explainability from Grad-CAM and attention visualization generate positive results that confirm the effectiveness of the proposed approach in real-world scenarios.

**VI. CONCLUSION**

To the best of our knowledge, this is the first research study that proposes a novel multimodal classification approach that utilizes explainable deep learning to classify onion services based on the image and text content of each site. This approach consists of two learning pathways, 1) image modality based on CNN with explainability using Grad-CAM and 2) text modality based on pre-trained word embedding and Bi-LSTMs with Bahdanau additive attention mechanism for explainability. We evaluated this approach on a state-of-the-art onion services dataset curated by CIRCL, containing more than 8000 instances of onion services across 51 categories.

The results of this experiment confirm the value and effectiveness of the proposed multimodal classification approach in enabling proactive CTI decision-making with interpretable and explainable outcomes to monitor and detect the cybersecurity threats originating from dark web onion services. Modularization of the proposed approach as an AI pipeline that can be plugged into threat analysis platforms like CIRCL’s AIL framework to enhance the cybersecurity threat shareability is a further contribution of this study. As future work, we intend to integrate further modalities like metadata, network access data, and other unstructured content recorded by an onion service. We will also explore how this approach can be made accessible to other CTI technology platforms. It is anticipated that proactive CTI methods such as this approach will become crucial in preventing cybercrimes and ensuring a safe and secure digitally connected future for human society.

**REFERENCES**

[1] J. Li, “Cyber security meets artificial intelligence: a survey,” Front. Inf. Technol. Electron. Eng., vol. 19, no. 12, pp. 1462–1474, 2018.
[2] A. Chetry and U. Sharma, “Dark web Activity on Tor: Investigation Challenges and Retrieval of Memory Artifacts,” in International Conference on Innovative Computing and Communications, 2021, pp. 953–964.
[3] P. Burda, C. Boot, and L. Alldén, “Characterizing the redundancy of DarkWeb onion services,” ACM Int. Conf. Proceeding Ser., 2019, doi: 10.1145/3339252.3339273.
[4] B. He, M. Patel, Z. Zhang, and K. C. C. Chang, “Accessing the deep web,” Commun. ACM, vol. 50, no. 5, pp. 94–101, 2007, doi: 10.1145/1230819.1241670.
[5] S. Ghosh, A. Das, P. Porras, V. Veyneswaran, and A. Gehani, “Automated categorization of onion sites for analyzing the darkweb ecosystem,” Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., vol. Part F1296, pp. 1793–1802, 2017, doi: 10.1145/3097938.3098193.
[6] J. Saleem, R. Islam, and M. A. Kabir, “The Anonymity of the Dark Web: A Survey,” IEEE Access, vol. 10, pp. 1–1, 2022, doi: 10.1109/access.2022.3161547.
[7] M. Ebrahimi, M. Surdeanu, and H. Chen, “Detecting cyber threats in non-english dark net markets: A cross-lingual transfer learning approach,” 2018 IEEE Int. Conf. Inf. Technl. Sensc. Informatics, IS 2018, pp. 85–90, 2018, doi: 10.1109/ISI2018.8587404.
[8] M. Mirea, V. Wang, and J. Jung, “The not so dark side of the darkest: a qualitative study,” Secur. J., vol. 32, no. 2, pp. 102–118, 2019, doi: 10.1057/s41284-018-0150-5.
[9] J. Bergman and O. B. Popov, “Journal of Digital Forensics , Security and Law The Digital Detective ‘s Discourse - A toolset for forensically sound collaborative digital forensics,” 2012 IEEE Int. Conf. Big Data, pp. 85–90, 2012, doi: 10.1109/ACCESS.2022.3176965, IEEE Access.
[26] A. Sun, E.-P. Lim, and W.-K. Ng, “Web classiﬁcation using support vector machine,” p. 96, 2002, doi: 10.1145/584931.584952. [44] H. Jiang et al., “A Multi-Label Deep Learning Model with Interpretable Grad-CAM for Diabetic Retinopathy Classiﬁcation,” Proc. Annul. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS, vol. 2020-July, pp. 1560–1563, 2020, doi: 10.1109/EMBC44109.2020.9175884.

[27] H. X. Xu, X. L. Hao, S. Y. Wang, and Y. F. Hu, “A method of deep Web classiﬁcation,” Proc. Sixth Int. Conf. Mach. Learn. Cybern. ICMLC 2007, vol. 7, no. August, pp. 4009–4014, 2007, doi: 10.1109/ICMLC.2007.4370847.

[28] M. Graczyk and K. Kinningham, “Automatic Product Categorization for Anonymous Marketplaces,” Comput. Sci., pp. 1–6, 2015.

[29] T. Sabbah, A. Selamat, M. H. Selamat, R. Ibrahim, and H. Fujita, “Hybridized term-weighting method for Dark Web classiﬁcation,” Neurocomputing, vol. 173, pp. 1908–1926, 2016, doi: 10.1016/j.neucom.2015.09.063.

[30] J. Dalins, C. Wilson, and M. Carman, “Criminal motivation on the dark web: A categorisation model for law enforcement,” Digital Investig., vol. 24, pp. 62–71, 2018, doi: 10.1016/j.dinin.2017.12.003.

[31] M. W. Al Nabki, E. Fidalgo, E. Alegre, and I. De Paz, “Classifying illegal activities on tor network based on web textual contents,” 15th Int. Conf. Eng. Chapter Assoc. Comput. Linguist. EACL 2017 - Proc. Conf., vol. 1, pp. 35–43, 2017.

[32] B. He, T. Tao, and K. C. C. Chang, “Organizing structured web sources by query schemes: A clustering approach,” Int. Conf. Inf. Knowl. Manag. Proc., pp. 22–31, 2004.

[33] I. D. Buldin and N. S. Ivanov, “Text Classiﬁcation of Illegal Activities on Onions Sites,” Proc. 2020 IEEE Conf. Russ. Young Res. Electr. Electron. Eng. EICON/Plus 2020, pp. 245–247, 2020, doi: 10.1016/ECONRus49466.2020.09.049.

[34] R. Biswas, E. Fidalgo, and E. Alegre, “Recognition of service domains on TOR dark net using perceptual hashing and image classiﬁcation techniques,” IET Semin. Dig., vol. 2017, no. 5, pp. 7–12, 2017, doi: 10.1094/i.c.2017.0041.

[35] E. Fidalgo, E. Alegre, V. González-Castro, and L. Fernández-Robles, “Illegal Activity Categorisation in DarkNet Based on Image Classiﬁcation Using CREIC Method,” in International Joint Conference SOCO ’17-CEIS’17-ICEUTE’17 León, Spain, September 6–8, 2017, Proceedings, 2018, pp. 600–609.

[36] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A Survey on Deep Transfer Learning,” in Artiﬁcial Neural Networks and Machine Learning -- ICANN 2018, 2018, pp. 270–279.

[37] B. Wang, Y. Sun, B. Xue, and M. Zhang, “Evolving deep convolutional neural networks by variable optimization for image classiﬁcation,” arXiv, vol. 24, no. 2, pp. 394–407, 2018.

[38] J. Gu et al., “Recent advances in convolutional neural networks,” Pattern Recognit., vol. 77, pp. 354–377, 2018, doi: 10.1016/j.patcog.2017.10.013.

[39] H. C. Shin et al., “Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning,” IEEE Trans. Med. Imaging, vol. 35, no. 5, pp. 1285–1298, 2016, doi: 10.1109/TMI.2016.2528162.

[40] A. Baldominos, Y. Saez, and P. Isasi, “Evolutionary convolutional neural networks: An application to handwriting recognition,” Neurocomputing, vol. 283, pp. 38–52, 2018, doi: 10.1016/j.neucom.2017.12.049.

[41] S. Chakraborty et al., “Interpretability of deep learning models: A survey of results,” 2017 IEEE SmartWorld Ubiquitous Intell. Comput. Adv. Trust. Comput. Scalable Comput. Commun. Cloud Big Data Comput. Internet People Smart City Innov. SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI 2017 - , pp. 1–6, 2018, doi: 10.1109/UIC-ATC.2017.8397411.

[42] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning Deep Features for Discriminative Localization,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 2921–2929, 2016, doi: 10.1109/CVPR.2016.319.

[43] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization,” Int. J. Comput. Vis., vol. 128, no. 2, pp. 336–359, 2020, doi: 10.1007/s11263-019-01228-7.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3176965, IEEE Access