Enhancing DarkWeb Activities Classification Using Embedding Methods

Meriem Sennad, Zineb Ellaky and Faouzia Benabbou

February 11, 2025

Enhancing DarkWeb Activities Classification Using Embedding Methods

Meriem Sennad
Hassan II University of Casablanca
Faculty of Sciences Ben M’Sick
Casablanca, Morocco
meriem.sennad-etu@etu.univh2c.ma

Zineb Ellaky
Hassan II University of Casablanca
Faculty of Sciences Ben M’Sick
Casablanca, Morocco
zinebellaky@gmail.com

Faouzia Benabbou
Hassan II University of Casablanca
Faculty of Sciences Ben M’Sick
Casablanca, Morocco
faouzia.benabbou@univh2c.ma

***Abstract*—The Dark Web is widely recognized for facilitating illicit activities such as drug trafficking, weapons trade, cybercrime, and hacking services. Tackling these activities poses a significant challenge for law enforcement and cybersecurity experts. Recent advancements in Artificial Intelligence (AI) and machine learning have shown great potential in various fields, including the analysis of the Dark Web. This article presents an innovative system designed to classify 10 types of illicit activities on the Dark Web. The system leverages the publicly available DUTA dataset, which provides a structured foundation for analyzing Dark Web content. Ensemble learning techniques, including Random Forest (RF), Light Gradient Boosting Machine (LightGBM), Extreme Gradient Boosting (XGB), Gradient Boosting (GB), and CatBoost, were employed to achieve this objective. To process textual data, three-word embedding techniques were utilized to convert text into vector representations. Notably, our approach demonstrates exceptional performance: the combination of the ELMo word embedding model and the XGB classifier achieved superior results, with an accuracy and precision rate of 99.95%. These findings highlight the effectiveness of our system in identifying and classifying illegal activities within the Dark Web ecosystem.**

Keywords— Dark web, Machine Learning, Classification, Illegal activities detection, Embedding, Balancing data, Security.

I. INTRODUCTION

The Dark Web originated in the late 1990s as a research initiative by the U.S. Department of Defense aimed at developing an encrypted network for secure communication. Initially designed to protect sensitive information for governments, businesses, universities, and research organizations, this network evolved into a private framework known as the "Internet Private." The project, later named Tor (The Onion Router), was released as open-source software, enabling global users to benefit from its privacy and security features. The anonymity provided by Tor has made the Dark Web a haven for privacy-conscious users, activists, and individuals seeking to bypass online censorship. Accessible only through specialized tools like Tor, Freenet, and I2P, the Dark Web represents a concealed portion of the internet that offers anonymity and encryption. While it provides legitimate users with a secure environment for privacy, it also facilitates illegal activities, raising significant concerns for cybersecurity and law enforcement.

The Dark Web hosts various activities, ranging from drug and weapon sales to the trade of stolen data, dissemination of illicit content, and discussions on sensitive topics. While some activities remain legal, others unequivocally cross into criminality, amplifying concerns about morality and safety. Furthermore, the Dark Web is a hub for cyber threats, including hacking, identity theft, espionage, and malicious software distribution. These activities often have real-world implications, demonstrating the urgent need for advanced analytical methods to address the associated risks. However, analyzing Dark Web data presents unique challenges. Datasets are frequently incomplete, imbalanced across activity categories, and difficult to process, complicating analysis and modeling efforts. This study addresses these challenges by developing an innovative system to detect and classify 10 illegal activities. Our approach leverages three advanced word embedding techniques to transform textual data into vectorized representations, enabling robust analysis and classification.

This study is structured into several key sections. The first provides an in-depth review of state-of-the-art classification techniques applied to the Dark Web. The second outlines the foundational concepts and frameworks that underpin our work. The third section details our proposed methodology, including the datasets, experimental setup, and validation processes used. Finally, we present and analyze the experimental results, offering insights into the effectiveness of our system and its contributions to enhancing online security and combating cybercrime.

II. RELATED LITERATURE

In a study conducted by Alaidi et al.[1], efforts were directed at collecting data to classify activities on the dark web. The research employed three data preprocessing techniques and applied three classification algorithms: Random Forest, Linear Support Vector Classifier (SVC), and Naïve Bayes. Among these, Linear SVC achieved the highest performance, with an accuracy of 91%, a precision of 89%, an F1-score of 88%, and a recall of 88%. Horasan et al. [2] concentrated on classifying network traffic using the CIC-Darknet2020 dataset. This study utilized several machine learning algorithms, including K-Nearest Neighbors (KNN), Logistic Regression (LR), Random Forest (RF), Support Vector Machines (SVM), Linear Discriminant Analysis (LDA), Extra Tree Classifier, and (GB). Among these, the GB algorithm demonstrated superior performance, achieving an accuracy of 99.8%, showcasing its effectiveness in classifying network traffic. Building on this theme, Rajawat et al. [3] explored the classification of criminal networks on the dark web using seven datasets. They evaluated three models: a Transductive

Support Vector Machine (TSVM), an SVM, and a hybrid Neural Network-Support Vector Machine (Fusion NN-SVM). Their results demonstrated that Fusion NN-SVM consistently outperformed the other algorithms across all datasets, showcasing its effectiveness in this domain. Zenebe et al. [4] collected a dataset of 14,865 instances, which was later reduced to 6,069 after preprocessing. Their study aimed to classify cyber threats from the dark web using three algorithms: Naïve Bayes (NB), Random Tree, and RF. RF emerged as the most effective, achieving an accuracy of 97.87%. Similarly, Rust-Nguyen et al. focused on darknet traffic classification and adversarial attacks [5]. The researchers utilized the CIC-Darknet2020 dataset for utilizing the CIC-Darknet2020 dataset. After evaluating eight algorithms, including KNN, SVM, Multilayer Perceptron, RF, GB, Extreme Gradient Boosting (XGB), Convolutional Neural Networks (CNN), and Generative Adversarial Networks (GAN), RF stood out with a remarkable F1-score of 99.8% for traffic classification. The main objective of Sarwar et al. in [6] addressed the detection and categorization of dark web activities using datasets such as DUTA-10K-GVIS and Anon17. By implementing preprocessing techniques like data balancing, they tested four algorithms: GB, RF, Decision Tree (DT), and XGB. Their findings showed that XGB achieved the best performance, with precision, recall, and F1-score rates of 85%. In another study, Demertzis et al.[7], utilized the CIC-Darknet2020 dataset to classify malicious traffic in the dark web. After evaluating 16 different algorithms, the Reservoir Model (RM) demonstrated the highest performance, achieving an accuracy of 94.51%, precision of 93.17%, recall of 81.22%, and an F1-score of 92.42%. Abu Al-Haija et al. [8] designed a darknet traffic detection system for IoT applications, testing several algorithms, including BAG-DT, ADA-DT, RUS-DT, O-DT, O-DSC, and O-KNN. Among these, BAG-DT showed the highest accuracy at 99.5%, highlighting its effectiveness in classifying darknet traffic for IoT applications. In another study, Cherqi et al. [9] collected dark web data to classify products in dark web marketplaces. They employed four algorithms: RF, SVC, NB, and LR, with NB demonstrating the best results, achieving precision, recall, and F1-score of 92%, 93%, and 93%, respectively. Sarkar et al. [10] worked on classifying TOR (The Onion Router) network traffic using the UNB-CIC dataset. They applied five models: DNN(B), DNN(A), Artificial Neural Network (ANN), ANN-SU, and ANN-CFS. DNN(B) achieved the best results, with an accuracy of 99.89%, precision of 99.88%, recall of 99.99%, and an F1-score of 99%. A related study [11] was to classify malicious and benign dark web content using the CIC-Darknet2020 dataset. Among the seven algorithms they tested, Random Forest achieved the highest accuracy. Tong et al. [12] examined the behavior of dark web traffic using the DIDARKNET dataset, exploring 13 machine learning algorithms. Their analysis showed that DARK-F performed the best with an accuracy of 87.84%, precision of 88.34%, recall of 87.84%, and an F1-score of 88.02%. Almomani et al. [13] aimed to classify darknet traffic using the CIC-Darknet2020 dataset and apply several machine learning algorithms, including NN, KNN, SVM, and Logistic LR. KNN achieved the highest performance with an accuracy of 94.8%, showcasing its effectiveness for darknet traffic classification. Thorat et al. [14] proposed a method to classify illegal activities on the dark web by using Naive Bayes, SVM, and Random Forest. The best result was achieved with Naive Bayes, which produced an accuracy of 88.89%. Alimoradi et al. [15] focused on darknet traffic classification to detect cyberattacks and malicious activities. Their study, which employed a deep neural network as the classifier, achieved an accuracy of 95% and a kappa score of 92%. Coutinho Marim et al. [16] used the CIC-Darknet dataset for darknet traffic detection, testing three models: DT, RF, and MLP. The MLP model showed the best performance with an accuracy of 99.92%, precision of 99.15%, and F1-score of 99.53%. In their study Zhang et al. [17] crawled data from drug-related groups on the Darknet to identify hazardous entities in the illegal drug domain. They applied three algorithms: Syntax-BERT-BiLSTM-CRF, BERT-BiLSTM-CRF, and BiLSTM-CRF. The best performance came from Syntax-BERT-BiLSTM-CRF, with a precision of 89%, recall of 79%, and F1-score of 84%.

Alshammery et Aljuboori, [18] classified illegal dark web activities, dividing them into five categories: Drugs, Others, Hacking, Fake ID, and Weapons. SVM performed the best, with an accuracy of 91%, precision of 89%, recall of 88%, and F1-score of 88%. In another study, Al Nabki et al. [19] used the DUTA dataset to classify activities on the dark web, utilizing various techniques such as TFIDF LR, BOW LR, and TFIDF NB. TFIDF LR provided the highest performance, with an accuracy of 96.6%. The work of Cascavilla et al. [20] combined several datasets, including Agora and Duta10k, to classify illicit content on the dark web using four models: BERT, LSTM, ULMFit, and RoBERTa. BERT was found to perform the best, with an accuracy of 96.08%, precision of 82%, recall of 78%, and F1-score of 80%. Pastor-Galindo et al. [21] used a hybrid approach for classifying terrorism activities, combining SD and UN functions with five algorithms: SVM, KNN, DT, NB, and ELM. The best result came from SVM with UN, achieving an accuracy and F1-score of 93%. In this study [22], Azhar et al. Focused on terrorism activities in the darknet using the LSTM model, which resulted in an accuracy and F1-score of 82%.

Al-Nabki et al. [23] used the DUTA-10K dataset to classify drugs on the dark web, testing ListNet, MLP, and RankNet. ListNet performed the best with an NDCG of 95%. Jin et al. [24] utilized the CODA dataset to classify dark web activities using BERT, SVM, and CNN. BERT achieved the best performance with a precision of 92.51%, recall of 92.50%, and F1-score of 92.49%. Li et al. [25] focused on classifying malicious content in the Tor darknet using KNN with TF-IDF and frequency-based features. KNN with TF-IDF yielded the best results, with a precision of 90%, recall of 89%, and F1-score of 89%. Finally, Kansagra et al. [26] proposed a system for classifying illegal activities on the dark web using the TOIC dataset and NB with TF-IDF, achieving an accuracy of 93.5%.

Table I provides a summary of the state of art approaches comparison based on the following elements:

- Objective: This means the objectives pursued by the authors.
- Dataset: the datasets utilized in the studies.
- Features type: type of the features or attributes in the dataset.
- Preprocessing: the preprocessing techniques

employed, such as data cleaning (DC), feature extraction (FE), Feature Selection (FS), Word Tokenization (WT), Remove stop words (RSW), Digitization of features (DF), Conversion of Timestamp (CT), Handling Missing Data (HMD), Data Balancing (DB), tag removal (TR), Data normalization (DN), Data shuffling (DS), Feature Encoding (FEN), removal of punctuation (PR), Number Removal (NR), etc.

- ML&DL: The research utilizes machine learning or deep learning techniques such as SVM, CNN, DT, NB, RF, etc. The models are ordered from the best to the least performing
- -performance: This refers to the performance obtained based on the following metrics: accuracy (A), precision (P), recall (R), and F1-score (F), as well as Normalized Discounted Cumulative Gain (NDCG).

TABLE I: THE COMPARISON TABLEE OF THE STATE OF THE ART APPROACHES

Ref.	Objective	Dataset	Preprocessing	ML&DL	Performance (%)
[1]	Classification activities	Dataset collected	DC, WT, RSW	SVC, RF, NB	A=91 ,P=89,F=88
[2]	Classification of network traffic	CICDarkNet2020	DF, CT	GB, RF, ETC, RNN, SVC, LR, LDA	A=99.8,
[3]	Classification of Criminal Networks in Dark Web	1-Android botnet dataset 2- CIC-InvesAndMal2019 3-CIC-DDoS2019 4- ISCXIDS2012 5- CIC-IDS2017 6- CSE-CIC-IDS2018 on AWS 7-CIRA-CIC-DoHBrw-2020	-	NN- S3VM, TSVM, SVM	A=81, A=65, A=61
[4]	Classification of the cyber threat from the dark web	Dataset collected	DC	RF, RT, NB	A=97
[5]	Classification of darknet traffic	CIC-Darknet2020	DC, DB	RF, CNN, SVM, AC-GAN, XGB, GBDT, k-NN, MLP	Tr F=99,8, APP F=92
[6]	detection and categorization of darknet	combined these two datasets (Anon17, DUTA)	HMD, DB, FS	XGB, DT, GB, RFR	P=85, R=85, F=85 P=85, R=84 F=84 P=83, R=81, F=81 P=71, R=81, F=81
[7]	darknet traffic analysis and network management framework	CIC-Darknet2020	-	RM , AutoKeras, Catboost, XGB, DT, RF, GB, ET, KNN, LGB, RIDGE, LD, QD, LR, NB, SVM	A=94.51,P=93.17,R=81.22,F=92.42
[8]	Detection System for IoT Applications	CIC-Darknet2020	FS, DN, DS	BAG-DT, ADA-DT RUS-DT, O-DT, O-DSC, O-KNN	A=99.5%
[9]	Classification of a product proposed in the Darkweb marketplace	Dataset collected	Words misspelling	NB, SVC, LR, RF	P=92, R=93, F=93
[10]	Classification of traffic in TOR	UNB-CIC	FS	DNN(B), DNN(A), ANN, ANN-SU, ANN-CFS	A=99.89,P=99.88,R=99.99,F=99
[11]	Classification the malicious and benign Dark Web content	CIC-Darknet2020		Rf, ET, DT, KNN, RIDGE, LR, MLP	RF had the highest performance.
[12]	Analysis of the Behavior of Dark Web Traffic	DIDarknet	-	DARK-F, DEEP-F, DEEP-I, RF, DT, KNN, MLP MLP-Atten, CNN, RBF-SVM, Linear-SVM, LR, NB	A=87.84,P=88.34,R=87.84,F=88.02
[13]	Classification of traffic darknet	CIC-Darknet2020	DC	KNN, LG, SVM, NN	A=92%
[14]	Classification and categorization of illegal activities on the dark web	Dataset collected	Feature extraction	NB, SVM, RF	A=87,R=87,F=88
[15]	Classification of Darknet Traffic	DIDarknet	-	NN	A=95, kappa=92%
[16]	detection and characterization of darknet traffic	CIC-Darknet2020	Labels correction, FEN, FE(one-hot-encoding)	MLP, RF, DT	P=99.92, R=99.15, F=99.53
[17]	Detection and investigation of Chinese public hazard entities in the Darknet	Dataset collected	-	Syntax-BERT-BiLSTM-CRF BERT-BiLSTM-CRF BiLSTM-CRF	P=0.89, R= 0.79, F= 0.84%

Ref.	Objective	Dataset	Preprocessing	ML&DL	Performance (%)
[18]	Classifying Illegal Activities in TOR	Dataset collected	TR, PR, elimination of numbers, tokenization, RSW	SVM NB	A=91, P=89, R=88, F=88
[19]	categorization of illegal activities in TOR	DUTA	FE (TFIDF(T), Bow(B))	T + LR, T + SVM B + LR B + SVM B + NB T + NB	A=96.6
[20]	Classification of Illicit Content in the dark web	Dataset collected	WT, SWR, LC, stemming	Bert, LSTM, ULMFit, RoBERTa	A=96.08, P=82, R=78, F=80
[21]	Classification of terrorism activities on the dark web	Dataset collected	DC	SVM, KNN, ELM, NB, DT	A=93,F=93,SD:A=93% , F=90%
[22]	Classification of cyber-terrorist in the dark web	Dataset collected	-	LSTM	A= 82, F=82
[23]	Classification of drugs in the dark web	DUTA-10K	FS	ListNet, MLP, RankNet	NDCG=95 , NDCG=71, NDCG=69
[24]	Classification of the activities on the dark web	CODA	-	BERT, SVM, CNN	P=92.51, R=92.50, F=92.49
[25]	Classification of Malicious Contents in Tor Darknet	DUTA-10k +Extracted Data	DC, FE(TF-IDF[T], Frequency[F])	T+KNN F+KNN	P=90, R=89, F=89
[26]	Classification of illegal activities in TOR.	TOIC	FE (TF-IDF)	NB	A=93.5

The literature review indicates that machine learning methods, particularly SVM, RF, NB, LR, and KNN, are the most widely applied techniques in classifying Dark Web activities, yielding significant results. These methods have been coupled with feature extraction approaches like embedding techniques and statistical methods such as TF-IDF, BOW, and BERT, which are highly effective for processing and representing textual data. The datasets used in these studies predominantly consist of data sourced from Tor, DUTA, CODA, and TOIC, which cover a wide array of Dark Web activities, providing a rich basis for analysis and classification.

This paper introduces a novel approach that combines three embedding techniques with ensemble machine learning methods to classify ten distinct illegal activities. While ensemble learning has demonstrated its potential in various classification tasks, its application in Dark Web activity classification remains an area with limited exploration.

III. METHODOLOGY

This paper presents a novel system designed to classify Dark Web activities using ensemble learning techniques to categorize ten illegal activities (Drugs, Counterfeit Credit Cards, Pornography, Hacking, Cryptolocker, Violence, Counterfeit Money, Counterfeit Personal Identification, Fraud, and Others.). As illustrated in Figure 1, the architecture of the proposed system is structured into several stages: 1) Data preparation, 2) Data preprocessing, 3) Feature embedding methods, 4) Data balancing via SMOTE-ENN, 5) Model construction, and 6) Model evaluation utilizing various performance metrics. The models employed in this study include RF, GB, XGB, LightGBM, and CatBoost, all of which have demonstrated strong performance in classification tasks. By incorporating these methods into our framework, we aim to enhance the accuracy and robustness of Dark Web activity classification.

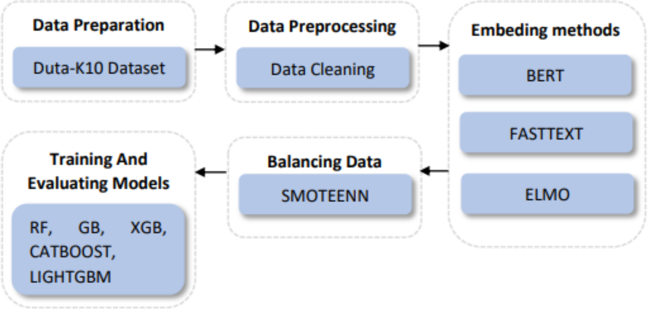

FIGURE 1: PROPOSED METHODOLOGY

A. Data Preparation

Use For model construction, we leveraged the Darknet Usage Text Addresses (Duta-10K) dataset, which includes activities from Dark Web pages and domains. This dataset, extracted from the Tor Hidden Service (HS) network, is accessible at [27]. The Duta-10K dataset consists of 10,367 Onion site entries, each associated with four key features and 28 distinct categories. These features are as follows:

- **Onion_Address**: This column identifies the Onion addresses, which are unique URLs within the Tor network. These addresses provide access to websites on the Dark Web and are key to maintaining anonymity and security in online browsing.
- **Main_Class**: This column assigns each Onion address to its primary category, representing the

main type of activities associated with the corresponding Dark Web site.

- **Sub_Class**: The sub-class column provides further granularity by indicating specific activities or operations related to each Onion address, giving a more detailed understanding of the Dark Web's diversity.
- **Lang**: This feature indicates the language of the content found on the website associated with each Onion address, offering insights into the geographic and linguistic distribution of Dark Web activities.

Since our study focuses on illegal activities, we performed a categorization of the Main_Class feature. This process aimed to group the most widely recognized illegal activities to organize our dataset better. By concentrating on these high-impact categories, we simplified the analysis, allowing for more straightforward identification of key trends and facilitating the development of focused strategies to address the most significant high-risk behaviors. As a result of this grouping, we identified 10 main categories of illicit activities: **Drugs**, **Counterfeit Credit Cards**, **Pornography**, **Hacking**, **Cryptolocker**, **Violence**, **Counterfeit Money**, **Counterfeit Personal Identification**, **Fraud**, and **Others**. The “Others” category also encompasses legal activities found on the Dark Web, including platforms for art, online casinos, digital libraries, and other services that, while legally compliant, utilize the Dark Web for enhanced privacy and anonymity. This categorization enables a more systematic analysis of illicit activities and allows us to focus on the most prominent and dangerous behaviors prevalent on the Dark Web.

B. Data Preprocessing

Data preprocessing plays a crucial role in ensuring the accuracy and reliability of results in data analysis and machine learning tasks. During this stage, we implemented various data cleaning procedures to enhance the dataset's quality, including addressing missing values and eliminating unnecessary spaces across all columns. For the *Onion_Address* column, we specifically removed the '.onion' suffix from the URLs, as it does not contribute valuable insights to our analysis. This preprocessing step ensures that the dataset is clean and consistent, which is vital for constructing effective models and achieving accurate classification outcomes.

C. Feature embedding methods

Embedding involves representing objects, words, phrases, or entities as vectors in a lower-dimensional space. The goal of embedding is to encode the semantic or structural relationships between these elements into a more compact and interpretable format. Various techniques are available to generate embeddings, each designed for specific data types and tasks [28]. Below are the three embedding methods utilized in this study, each selected for its suitability in representing the specific features of the data and enhancing the model's performance. 1) ELMo (Embeddings from Language Models): A model that generates contextual embeddings by leveraging context information in a sentence for each word [29]. 2) BERT (Bidirectional Encoder Representations from Transformers): [30] A language representation model that generates contextual embeddings using deep bidirectional Transformers, simultaneously capturing both left and right context in all layers. 3) FastText: An extension of Word2Vec that considers the morphological structure of words by breaking them into sub-words, allowing a more efficient representation of rare and compound words [31].

D. Balancing Data Using SMOTE-ENN

After preparing the data, we encountered an issue with class imbalance, as our dataset contained 10 classes of unequal sizes. This imbalance could result in the model favoring the majority classes and neglecting the minority during training. To mitigate this issue, we applied the SMOTE-ENN technique, which combines two methods: SMOTE (Synthetic Minority Over-sampling Technique) and ENN (Edited Nearest Neighbors). SMOTE works by generating synthetic data points for the minority class, thereby enlarging its representation. Conversely, ENN cleans up the majority class by removing redundant or noisy data points [28]. This hybrid approach ensures a more balanced class distribution by augmenting the minority classes and refining the majority class, ultimately improving the model's ability to distinguish between classes. The distribution of the Main_Class before and after applying SMOTE-ENN is shown in Table II below.

TABLE II: MAIN_CLASS DISTRIBUTION BEFORE AND AFTER SAMPLING

Main_Class	Count before SMOTE-ENN	Count after SMOTE-ENN
Drugs	465	8590
Counterfeit Credit-Cards	399	8590
Porno	253	8590
Hacking	205	8590
Cryptolocker	185	8590
Violence	95	8590
Counterfeit Money	83	8590
Counterfeit Personal-Identification	72	8590
Fraud	20	8590
Others	8590	8590

E. Ensemble methods

Ensemble methods represent a class of machine learning techniques that combine predictions from multiple weaker models to enhance accuracy, robustness, and overall model reliability. Prominent ensemble techniques include bagging methods such as RF, boosting algorithms like GB, XGBoost, CatBoost, and LightGBM, and stacking, which integrates outputs from various models into a cohesive framework. Cross-validation was implemented to validate these models and mitigate overfitting, enabling a comprehensive evaluation of model performance by partitioning the dataset into multiple training and testing subsets. Below is a concise summary of the models used in this study: **1) Random Forest:** A tree-based ensemble method that builds multiple decision trees using randomly selected subsets of features [32]. This approach minimizes variance and enhances generalization, making it effective for classification and regression tasks. **2) Gradient Boosting**: This method incrementally develops models by addressing errors made in previous iterations. It excels in handling complex classification and regression problems by optimizing predictive performance step by step [33]. **3) XGBoost** (eXtreme Gradient Boosting): An advanced version of Gradient Boosting, XGBoost integrates regularization techniques, tree pruning, and adjustable learning rates to improve performance and reduce overfitting [33]. It also efficiently manages missing values, ensuring reliability across diverse datasets. **4) CatBoost:** Explicitly designed for datasets with categorical variables, CatBoost eliminates the need for extensive preprocessing [34]. It employs a unique "ordered boosting" mechanism to enhance optimization and incorporates robust regularization methods to prevent overfitting. This algorithm performs exceptionally well on small and large datasets with categorical features. **5) LightGBM** (Light Gradient Boosting Machine): Known for its speed and efficiency, LightGBM uses a 'leaf-wise' tree growth strategy, selecting splits based on the highest loss reduction. This leads to more balanced trees and faster training times than the 'level-wise' method in other boosting algorithms like XGBoost. It is particularly well-suited for large datasets and complex predictive tasks.

IV. Results and Discussion

A. Performance metrics

To assess the effectiveness of our system, we employed key evaluation metrics: accuracy, precision, recall, and F1-score. These metrics were calculated using four fundamental parameters: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). Each metric serves a specific purpose in evaluating different aspects of the model's performance:

Accuracy: Represents the proportion of correct predictions over the total predictions made by the model. It is calculated as:

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

Precision: Indicates the proportion of correctly predicted positive instances among all predicted positives. It is given by:

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

Recall (Sensitivity): Measures the model's ability to identify all relevant positive instances. It is defined as:

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

F1-Score: Provides a harmonic mean of precision and recall, offering a balanced evaluation considering both FP and FN. It is calculated as:

$$F1\text{-}Score = \frac{2 * (Precision * Recall)}{(Precision + Recall)}$$

B. Approach evaluation and comparaison

In the experiment, we used a machine running Windows 10 with a CORE i7 10th-gen processor, 12 GB of RAM, and a 1,000 GB hard drive. We implemented our model using Keras (2.2.4), Tensorflow (1.14.0), Python (Version 3.7.3), and Jupyter (6.0.3).

This section details the outcomes achieved by implementing the ensemble machine learning methods introduced earlier. The evaluation metrics have been computed and are summarized in Table II. These results offer a comparative analysis of the various methods, highlighting the strengths and limitations of each in accurately classifying activities on the Dark Web. The table provides a holistic view of the models' effectiveness, aiding in the identification of the most suitable ensemble approach for this classification task.

TABLE III: RESULTS ACHIEVED BY THE MODELS

Algorithm	**Methods**	**Train**				**Test**			
		Accuracy	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
RF	ELMO	99,82%	99,83%	99,57%	99,70%	99,82%	99,83%	99,57%	99,79%
	FastText	99,84%	99,85%	99,63%	99,74%	99,82%	99,83%	99,57%	99,70%
	Bert	97,54%	97,46%	95,90%	96,30%	97,52%	97,46%	95,89%	96,27%
GB	ELMO	85,74%	84,99%	85,57%	85,53%	85,72%	84,97%	85,52%	85,53%
	FastText	91,75	91,75	87,54	91,27	91,72	91,73	87,54	91,26
	Bert	83,23	82,99	81,90	81,91	83,22	82,97	81,88	81,89
XGB	ELMO	99,95%	99,95%	99,89%	99,92%	99,95%	99,95%	99,89%	99,92%
	Fast Text	99,89%	99,89%	99,74%	99,81%	99,85%	99,86%	99,72%	99,79%
	Bert	99,83%	99,83%	99,63%	99,73%	99,83%	99,83%	99,61%	99,70%
LIGHTGBM	ELMO	99,94%	99,93%	99,90%	99,94%	99,92%	99,91%	99,89%	99,92%
	FastText	99,89%	99,89%	99,75%	99,82%	99,86%	99,87%	99,72%	99,80%
	Bert	99,83%	99,83%	99,63%	99,73%	99,83%	99,82%	99,63%	99,80%
Catboost	Elmo	98,57%	98,57%	98,55%	98,53%	98,56%	98,56%	98,53%	98,54%
	FastText	93,50%	93,98%	93,80%	93,63%	93,47%	93,95%	93,79%	93,61%
	Bert	92,79%	92,79%	92,83%	92,54%	92,78%	92,76%	92,81%	92,53%

From the analysis, XGBoost combined with ELMo emerges as the best-performing model for this classification task, achieving exceptional metrics across both cross-validation and test datasets. The model attained Accuracy (99.95%), Precision (99.95%), Recall (99.89%), and F1-Score (99.92%), demonstrating its superior ability to classify Dark Web activities correctly. Notably, the consistency between the cross-validation and test results highlights the model's robustness and effective generalization to unseen data. LightGBM and ELMo closely followed, delivering similarly high metrics (99.94% accuracy) and confirming its reliability. Random Forest with FastText also demonstrated strong performance (99.84% accuracy) but slightly lagged behind XGBoost and LightGBM. CatBoost with ELMo, while achieving solid results (98.57% accuracy), was comparatively less effective.

The analysis highlights the critical role of embeddings like ELMo, FastText, and BERT, with ELMo consistently driving superior results. Using SMOTE-ENN further ensured balanced class distributions, enabling high precision and recall even for minority classes. Combining ensemble methods, advanced embeddings, and data balancing techniques proved pivotal for accurate Dark Web activity classification.

V. Conclusion

This study evaluates the performance of various ensemble methods, including RF, GB, XGBoost, LightGBM, and CatBoost, in classifying illicit activities on the Dark Web. Among the models, XGBoost paired with ELMo embeddings demonstrated the highest effectiveness, achieving outstanding metrics such as 99.95% accuracy, 99.95% precision, and 99.89% recall. The results underscore the significant role of ELMo in feature extraction, outperforming other techniques in capturing the nuances of the dataset and contributing to the superior performance of the ensemble methods.

For future research, we aim to broaden the scope by incorporating additional datasets, potentially obtained through web scraping or synthetically generated, to test our models' robustness and adaptability. Efforts will also focus on overcoming scalability challenges to ensure the proposed methods can handle more extensive and more complex datasets. Additionally, we plan to incorporate advanced deep learning techniques, particularly transformer-based models, to enhance the accuracy and efficiency of our methods. Another avenue for exploration involves addressing ethical concerns and developing frameworks to ensure the responsible application of these methodologies. By engaging in these initiatives, we aspire to contribute to the broader goal of advancing automated systems for identifying and combating illicit activities in Dark Web ecosystems.

References

[1] Alaidi AHM, Al_airaji RM, Alrikabi HThS, Aljazaery IA, Abbood SH. Dark Web Illegal Activities Crawling and Classifying Using Data Mining Techniques. Int J Interact Mob Technol 2022;16:122–39. https://doi.org/10.3991/ijim.v16i10.30209.

[2] Horasan F, Yurttakal AH. Gradyan Artırma Algoritması ile Karanlık Ağ Web Trafiği Sınıflandırması. IJERAD 2022;14:794–8. https://doi.org/10.29137/umagd.1117634.

[3] Rajawat AS, Bedi P, Goyal SB, Kautish S, Xihua Z, Aljuaid H, et al. Dark Web Data Classification Using Neural Network. Computational Intelligence and Neuroscience 2022;2022:1–11. https://doi.org/10.1155/2022/8393318.

[4] Zenebe A, Shumba M, Carillo A, Cuenca S. Cyber Threat Discovery from Dark Web, n.d., p. 174–163. https://doi.org/10.29007/nkfk.

[5] Rust-Nguyen N, Sharma S, Stamp M. Darknet traffic classification and adversarial attacks using machine learning. Computers & Security 2023;127:103098. https://doi.org/10.1016/j.cose.2023.103098.

[6] Sarwar MB, Hanif MK, Talib R, Younas M, Sarwar MU. DarkDetect: Darknet Traffic Detection and Categorization Using Modified Convolution-Long Short-Term Memory. IEEE Access 2021;9:113705–13. https://doi.org/10.1109/ACCESS.2021.3105000.

[7] Demertzis K, Tsiknas K, Takezis D, Skianis C, Iliadis L. Darknet Traffic Big-Data Analysis and Network Management for Real-Time Automating of the Malicious Intent Detection Process by a Weight Agnostic Neural Networks Framework. Electronics 2021;10:781. https://doi.org/10.3390/electronics10070781.

[8] Abu Al-Haija Q, Krichen M, Abu Elhaija W. Machine-Learning-Based Darknet Traffic Detection System for IoT Applications. Electronics 2022;11:556. https://doi.org/10.3390/electronics11040556.

[9] Cherqi O, Mezzour G, Ghogho M, El Koutbi M. Analysis of Hacking Related Trade in the Darkweb. 2018 IEEE International Conference on Intelligence and Security Informatics (ISI), Miami, FL: IEEE; 2018, p. 79–84. https://doi.org/10.1109/ISI.2018.8587311.

[10] Sarkar D, Vinod P, Yerima SY. Detection of Tor Traffic using Deep Learning. 2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA), Antalya, Turkey: IEEE; 2020, p. 1–8. https://doi.org/10.1109/AICCSA50499.2020.9316533.

[11] Allhusen A, Al-Ramahi M, Alsmadi I, Wahbeh A, Al-Omari A. Dark Web Analytics : A Comparative Study of Feature Selection and Prediction Algorithms n.d.

[12] Tong X, Zhang C, Wang J, Zhao Z, Liu Z. Dark-Forest: Analysis on the Behavior of Dark Web Traffic via DeepForest and PSO Algorithm. Computer Modeling in Engineering & Sciences 2023;135:561–81. https://doi.org/10.32604/cmes.2022.022495.

[13] Almomani A. Darknet traffic analysis, and classification system based on modified stacking ensemble learning algorithms. Inf Syst E-Bus Manage 2023. https://doi.org/10.1007/s10257-023-00626-2.

[14] Thorat H, Thakur S, Yadav A. Categorization of Illegal Activities on Dark Web using Classification 2020;07.

[15] Alimoradi M, Zabihimayvan M, Daliri A, Sledzik R, Sadeghi R. Deep Neural Classification of Darknet Traffic. In: Cortés A, Grimaldo F, Flaminio T, editors. Frontiers in Artificial Intelligence and Applications, IOS Press; 2022. https://doi.org/10.3233/FAIA220323.

[16] Coutinho Marim M, Ramos PVB, Vieira AB, Galletta A, Villari M, De Oliveira RM, et al. Darknet traffic detection and characterization with models based on decision trees and neural networks. Intelligent Systems with Applications 2023;18:200199. https://doi.org/10.1016/j.iswa.2023.200199.

[17] Zhang P, Wang X, Ya J, Zhao J, Liu T, Shi J. Darknet Public Hazard Entity Recognition Based on Deep Learning. Proceedings of the 2021 ACM International Conference on Intelligent Computing and its Emerging Applications, Jinan China: ACM; 2021, p. 94–100. https://doi.org/10.1145/3491396.3506525.

[18] Alshammery MK, Aljuboori AF. Classifying Illegal Activities on Tor Network using Hybrid Technique. Eijs 2022:3994–4004. https://doi.org/10.24996/ijs.2022.63.9.30.

[19] Al Nabki MW, Fidalgo E, Alegre E, De Paz I. Classifying Illegal Activities on Tor Network Based on Web Textual Contents. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, Valencia, Spain: Association for

Computational Linguistics; 2017, p. 35–43. https://doi.org/10.18653/v1/E17-1004.

[20] Cascavilla G, Catolino G, Sangiovanni M. Illicit Darkweb Classification via Natural-language Processing: Classifying Illicit Content of Webpages based on Textual Information: Proceedings of the 19th International Conference on Security and Cryptography, Lisbon, Portugal: SCITEPRESS - Science and Technology Publications; 2022, p. 620–6. https://doi.org/10.5220/0011298600003283.

[21] Pastor-Galindo J, Gómez Mármol F, Martínez Pérez G. On the gathering of Tor onion addresses. Future Generation Computer Systems 2023;145:12–26. https://doi.org/10.1016/j.future.2023.02.024.

[22] Azhar SN, Zolkipli MF. Exploring and Evaluating a Malicious Site on the Dark Web Using Machine Learning n.d.;6.

[23] Al-Nabki MW, Fidalgo E, Alegre E, Chaves D. Content-Based Features to Rank Influential Hidden Services of the Tor Darknet 2019. https://doi.org/10.48550/arXiv.1910.02332.

[24] Jin Y, Jang E, Lee Y, Shin S, Chung J-W. Shedding New Light on the Language of the Dark Web. Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Seattle, United States: Association for Computational Linguistics; 2022, p. 5621–37. https://doi.org/10.18653/v1/2022.naacl-main.412.

[25] Li R, Chen S, Yang J, Luo E. Edge-Based Detection and Classification of Malicious Contents in Tor Darknet Using Machine Learning. Mobile Information Systems 2021;2021:1–13. https://doi.org/10.1155/2021/8072779.

[26] Kansagra Z, Ahire K, Mishra P, Sarkar A. Classification of Illicit Venture on Dark Web:- A Survey 2020;5.

[27] Datasets – DUTA-10K – GVIS n.d. https://gvis.unileon.es/datasets-duta-10k/ (accessed October 5, 2024).

[28] Ellaky Z, Benabbou F. Political social media bot detection: Unveiling cutting-edge feature selection and engineering strategies in machine learning model development. Scientific African 2024;25:e02269. https://doi.org/10.1016/j.sciaf.2024.e02269.

[29] Kutuzov et Kuzmenko - 2021 - Representing ELMo embeddings as two-dimensional te.pdf n.d.

[30] Sazan SA, Miraz MH, Rahman ABMM. Enhancing Depressive Post Detection in Bangla: A Comparative Study of TF-IDF, BERT and FastText Embeddings. AETiC 2024;8:34–50. https://doi.org/10.33166/AETiC.2024.03.003.

[31] Ellaky Z, Benabbou F, Ouahabi S, Sael N. Word Embedding for Social Bot Detection Systems. 2021 Fifth International Conference On Intelligent Computing in Data Sciences (ICDS), Fez, Morocco: IEEE; 2021, p. 1–8. https://doi.org/10.1109/ICDS53782.2021.9626752.

[32] Liu J. Suppression of polarization random noise in a two-dimensional force sensorbased on random forest. J Sens Sens Syst 2025;14:1–11. https://doi.org/10.5194/jsss-14-1-2025.

[33] Razavizadeh NT, Salari M, Jafari M, Sabaghian E, Ghavami V. Comparison of Two Methods, Gradient Boosting and Extreme Gradient Boosting to Pre- dict Survival in Covid-19 Data. Jbe 2024. https://doi.org/10.18502/jbe.v9i3.15450.

[34] Hadianto A, Utomo WH. CatBoost Optimization Using Recursive Feature Elimination. Join 2024;9:169–78. https://doi.org/10.15575/join.v9i2.1324.