



Darknet traffic analysis, and classification system based on modified stacking ensemble learning algorithms

Ammar Almomani^{1,2}

Received: 25 April 2022 / Revised: 31 July 2022 / Accepted: 2 August 2022

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract

Darknet, a source of cyber intelligence, refers to the internet's unused address space, which people do not expect to interact with their computers. The establishment of security requires analyses of the threats characterizing the network. New machine learning classifiers known as stacking ensemble learning are proposed in this paper to analyze and classify darknet traffic. In dealing with darknet attack problems, this new system uses predictions formed by 3 base learning techniques. The system was tested on a dataset comprising more than 141,000 records analyzed from CIC-Darknet 2020. The experiment results demonstrated the study's classifiers' ability to distinguish between the malignant traffic and benign traffic easily. The classifiers can effectively detect known and unknown threats with high precision and accuracy greater than 99% in the training and 97% in the testing phases, with increments ranging from 4 to 64% by current algorithms. As a result, the proposed system becomes more robust and accurate as data grows. Also, the proposed system has the best standard deviation compared with current A.I. algorithms.

Keywords Ensemble learning · Machine learning · Darknet traffic analysis · Artificial intelligence classifiers · Virtual private network · Tor browser

1 Introduction

A darknet/dark web refers to a portion of I.P. space that's allocated and routed. In these spaces, inactive services and servers are located. It may include systems that are invisible and aimed at receiving messages. Such systems do not show a response

✉ Ammar Almomani
ammarnav6@bau.edu.jo; ammar.almomani@skylineuniversity.ac.ae

¹ IT Department, Al-Huson University College, Al-Balqa Applied University, P. O. Box 50, Irbid, Jordan

² School of Information Technology, Skyline University College, P.O. Box 1797, Sharjah, UAE

to anything, and they may be a part of an overlay network. Communication non-standard ports and protocols can be used to access this network.

Regarding the traffic that is destined to the darkness, it's deemed suspicious. It can be malicious, and it can be a misconfiguration as well. The systems used for monitoring can be set up in darknets to have the trap attackers attracted for intelligence gathering. Botnets (Al-Nawasrah et al. 2018; Alieyan et al. 2018) and malware (Al-Kasassbeh et al. 2020) often lack intelligence. Unfortunately, some people utilize unethical methods to increase their link count and reputation, such as propagating fake news via texts, photographs, and videos (Sahoo and Gupta 2021).

A darknet may be called black hole monitors, dark space, network telescopes (Pang et al. 2004), and spurious traffic. In addition to being smaller than real traffic, darknet traffic contains malicious activity traces. It is beneficial to analyze this traffic to identify the trend of attacks in the real network. Each attack has a specific way of exploiting the data existing in the network. Identifying those patterns facilitates tracing them to their corresponding attacks and clustering. It has been deemed effective for identifying the patterns in unclassified data, such as darknet traffic.

Many research questions will be discussed in our research such as, What are the darknet and Cyberspace? What is the Tor? Who uses Tor? Is the Dark Web a seedy place for "bad guys"?, Is it possible to determine the paths of the darknet, and what is the way to do that? Are AI algorithms best suited for identifying darknet paths?

Cyberspace is evolving into a more complex one, and so are the security dynamics. As our reliance on cyberspace has increased throughout time, monitoring cyberspace security systematically and regularly is increasingly crucial. A darknet system is a monitoring system that aims to detect activities deemed malicious and cyberspace's attack patterns. Meanwhile, darknet traffic is spurious traffic observed within the empty address space (i.e. a set of globally valid (I.P.) addresses assigned to any device or host (Niranjana et al. 2020)).

The use of darknets has increased following the outbreak of the COVID-19 pandemic, including among criminals with no prior cyber experience. Currently, COVID-19-related merchandise is being sold on darknet forums that have been previously related to narcotics. As a result, more and more new users seek advice from experienced darknet criminals on criminal opportunities, resulting in increased threats (UNODC 2021).

Cybercrime is projected to cost the world \$10.5 trillion annually by 2025, from \$3 trillion in 2015, according to a new report (Morgan 2021). According to Cybersecurity Ventures, global cybercrime expenditures are expected to climb by 15% over the next five years. In addition to being the most significant transfer of economic wealth in history, in the long run, this will harm the incentives for innovation and investment, and it will be more profitable than the worldwide trade in all major illegal drugs combined (Morgan 2021). An estimated 5000-fold increase in the size of the darknet web attack (which cannot be indexed or found by search engines) and a growth rate that defies quantification (Morgan 2021) have been reported. These reasons motivated us to employ deep representation learning based on stacking simple learning to address these challenges (Morgan 2021).

The ensemble methods attempt to construct several hypotheses and have them combined. Usually, the approach yields results that are deemed better than the

results reached by using a single strategy. This approach offers better generalizations and an improved ability to escape from local optima. It also offers excellent and advanced search potential. In this paper, we offered an ensemble scheme that's novel and based on two main layers, namely a base module and a combining module. The proposed system aims to employ the base modules and combine ones to distinguish easily between darknet traffic and benign traffic. Although some authors have done studies to analyze the Dark Web, there has yet a thorough literature review on the Dark Web's evaluation in the context of risks, which has motivated us to make our proposal.

The main contribution and novelty: New machine learning classifiers called stacking ensembles are proposed in this paper to analyze and classify darknet traffic. Which is used for the first time in Darknet attack problems. Based on this study's experimental results, the classifiers can easily distinguish between malignant traffic and benign traffic. They can detect unknown threats effectively by showing *that are greater than 99% in the training phase and 97% in the testing phase, with increments ranging from 4%-64% by current algorithms and 93% precision.* However, our system proves to have the best results compared with other algorithms used in the same area of research. Also, our system will adapt 3 reasons behind the effectiveness and success of ensemble learning in ML: a statistical reason, computational nature, and a representational reason, which will be discussed in Sect. 2. Moreover, the proposed system becomes stronger and more accurate as data grows. Also, the proposed system has the best standard deviation compared with current A.I. algorithms.

Section 2 discusses the Background and related work that includes ensemble learning background, which discusses machine learning using various methods to create hybrid approaches. Section 2 reviews the Background and related works for darknet and ensemble learning. Section 3 proposes a methodology for outlining the new darknet traffic analysis and classification system using the stacking ensemble learning classifiers approach. Section 4 comprises the experiments and results and describes the data used in this paper. Finally, Sect. 5 includes the conclusions and implications for future work.

2 Background and related work

This section includes two parts of the discussion: ensemble learning background and related work.

2.1 Darknet and ensemble learning background

Most of us can only view a fraction of what is on the dark web/darknet because Google's search results are limited. A search engine like Google, Yahoo!, or Bing can only index 5% of the web, and here is where most people stay. There is a term for the Internet's furthest reaches: the deep web. Search engines do not index the deep web since transitory websites appear and disappear, and their content does not follow the same regulations (Gdata 2022).

You'll need to download particular browsers to access the dark web, which is governed by no rules. On the dark web, websites like Silk Road, the most well-known example, sell illegal goods, including pirated films and other illicit substances, firearms, and pornographic material. Recent research has made it more challenging for criminals to hide in this location (Gdata 2022). Accordingly, Fig. 1 shows how to access the deep/dark web/darknet based on the Tor browser scenario.

Ensemble and hybrid ML is the most accurate and trustworthy ML algorithm. It is possible to develop hybrid ML models by combining ML techniques with other ML techniques or optimizing soft computing strategies. The ensemble methods are developed by employing several grouping methods, like the boosting or bagging methods for using several ML classifiers (Ardabili et al. 2019).

In recent years, machine learning has been incorporating many different techniques to create hybrid systems. These ensemble approaches to issue classification are often known as many classifier systems. (Du et al. 2012; Woźniak et al. 2014). There are two types of ensemble methods: similar and serial (or concatenation). Approaches are merged consecutively in a serial combination. For each subsequent analysis, the initial results are used as inputs. (Ponti 2011). It is the output created by the series' final method that is used to determine the final output (estimate, classification), as well as dimension reduction, clustering, and so on. Multiple data collection methods are used in parallel as part of the approach's parallel combination.

The decision rule is used to determine the final result. (Ponti 2011) as shown in Fig. 2.

Figure 2 shows the generally employed two-level. Through the 1st level, a group of several base learners shall be obtained from specific training data. Through the 2nd level, the learners obtained in the former phase shall be combined to generate

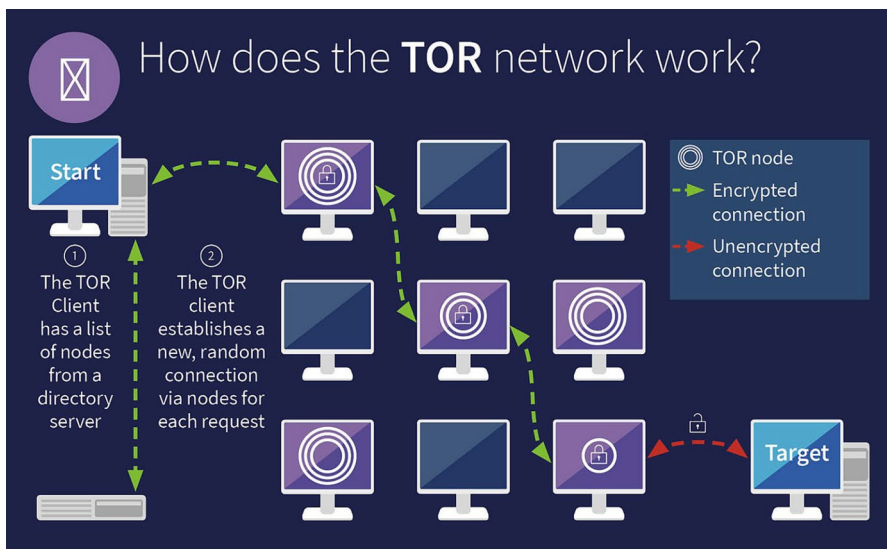


Fig. 1 How to access the deep/darkweb/darknet based on the Tor browser scenario (Gdata 2022)

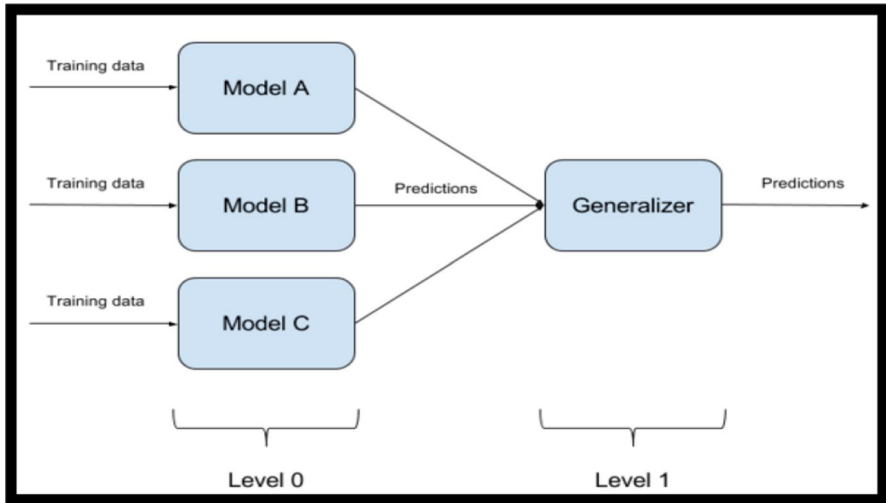


Fig. 2 The basic architecture of ensemble classifiers

a unified prediction model. The forecasts based on several base learners are developed and combined into a composite enhanced model superior to the individual base models. Having all the good individual models integrated into 1 composite enhanced model shall raise the accuracy level.

Ensemble models have become more effective in recent years, attributed to their high performance in various tasks, including addressing regression and classification-related problems (Divina et al. 2018). Ensemble methods comprise various learning models to improve the results obtained by each model. Ensemble learning was first examined in the 90 s (Hansen and Salamon 1990; Perrone and Cooper 1992). It has been proven that learning multiple and weak algorithms could be turned into strong ones. In a nutshell, ensemble learning (Dietterich 2000) is a measure through which multiple learner modules are implemented on a specific dataset to extract multiple predictions, which are then combined into 1 prediction to become a composite prediction.

Based on ref. (Dietterich 2000), there are 3 reasons behind the effectiveness and success of ensemble learning in ML. Regarding the 1st reason, it is a statistical reason. Models search for hypothesis space H to identify the hypothesis that's deemed the best. Since the datasets are usually limited, one is capable of funding numerous various hypotheses in H . The 2nd reason has a computational nature. Numerous models operate by making a local search operation to minimize the error functions. Those search operations may get stuck within the optima that's local. An ensemble created through initiating the local search operation from various points shall lead to an enhanced approximation of the unknown and true function. The 3rd reason, it's a representational reason. The unknown function that the research is searching for may not be included in H in numerous situations. Despite that, a combination of various hypotheses derived from H can enlarge the space of representable functions that may include a true function that's unknown (Divina et al. 2018). The basic

ensemble methods that are known and used the most are the bagging, boosting, and stacking methods.

Through bagging in that scheme, several models were developed. Regarding the results generated through these models, they are deemed equal. A voting mechanism shall be employed for settling on the majority result. If there is a regression, the average predictions are represented usually in the final output, as shown in Fig. 3.

Boosting method is comparable to the bagging method, except that it includes 1 modification of a conceptual nature. Instead of having weights that are equally assigned to models, the boosting method shall have various weights assigned to the concerned classifiers. It shall derive its final result in a manner that is based on weighted voting. If regression is present, a weighted average will usually be represented in the final output, as shown in Fig. 4.

Regarding the stacking method, it creates its models by employing several learning algorithms and a combiner algorithm, whereby the latter is trained to make the ultimate predictions using base algorithms, as shown in Fig. 5.

The ensemble model was built using three machines that learned from a variety of different categories in our research. There were three types of statistical machine learning: random forest (R.F.), support vector machines, and neural networks as computational machine learning (SVM). The details will be provided in section four as the proposed methodology.

2.2 Related works

The emergence of artificial Intelligence (A.I.) and ML-based technologies has led to the development of systems to detect threats. The ones who launched the attack adapted using A.I. as a tool deemed offensive. Several scholars have carried out works on darknet monitoring to have the threats detected. Such threats include botnets and DDoS. Regarding the feature sets employed in their research, they were

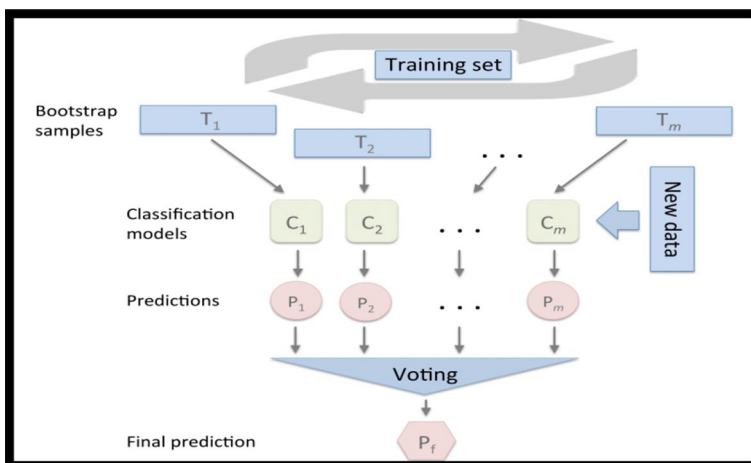


Fig. 3 Bagging—building an ensemble of classifiers from bootstrap samples (Oreilly 2022)

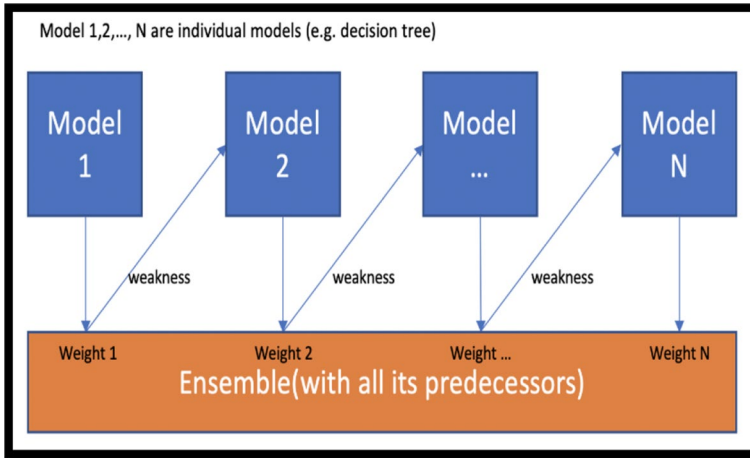


Fig. 4 Boosting algorithm developed for classification problems (Zhang 2019)

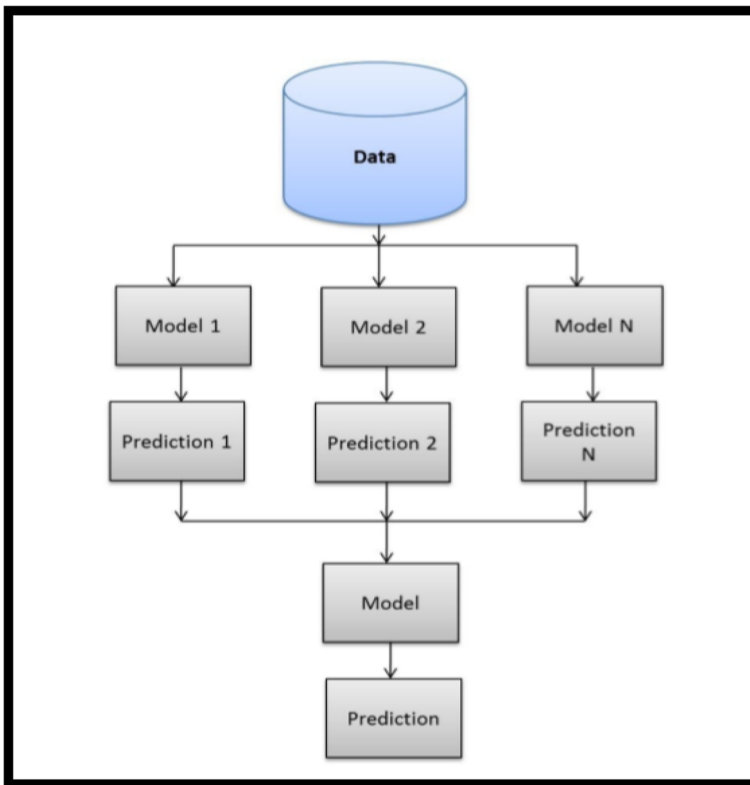


Fig. 5 Stacking generalization (Patel 2020)

small. They were limited to identifying a specific type of attack. In their study, Patel et al. (Bou-Harb et al. 2016) created a CSC-Detector. This detector was employed to have probings of large scale detected. The latter researchers focus on detecting the fingerprinting methods and activities instead of identifying the fingerprinting sources. They employed 250 GB of real darknet data for making trials. Zhang et al. (2017) proposed UnitecDEAMP to have the malicious events in darknet traffic profiled. They categorized and segmented the flows extracted from darknet traffic based on assessments of behavioral nature. They were capable of detecting malicious events that are significant.

Studies and articles on the analysis of darknet traffic have been increasing, and the reviewed studies provide the researcher with data about how darknet traffic may be employed to analyze security. For instance, Wang et al. (2011) employed darknet traffic to infer the temporal internet worm behaviors by implementing statistical nature methods for making an estimation. Maximum likelihood, methods of moments, and linear regression estimators were used in their study. In another study, Dainotti et al. (2014) displayed the measurement and carried out a horizontal scan-based analysis for the whole IPv4 darknet. They shed light on the space used in 2011 by the Sality botnet by having the botnet behavior visualized, correlated, and extrapolated across the internet, using general methods.

Bou-Harb et al. (2017) offered a unique probabilistic and preprocessing darknet model, with the ability to sanitize data and reduce the dimensions of big data using the extraction and analysis of probing time series. Bou-Harb et al. (2016) leveraged darknet data that are unsolicited and real. They created a system called (the CSC-Detector system) to identify the Campaigns launched for cyber scanning. The latter researchers empirically validated and evaluated the system using 240 GB of real darknet data. Regarding the outcome, it has disclosed 3 recent and probing large-scale campaigns. Such campaigns target various internet services.

Bou-Harb et al. (2017) examined data sanitization and cyber situational awareness by analyzing 910 GB of real Internet-scale traffic passively gathered through monitoring approximately 16.5 million darknet I.P. addresses. The authors sanitized the darknet data so that the data could be employed effectively in the operation of cyber threat intelligence generation. Meanwhile, Cambiaso et al. (2019) reviewed the literature on attacks launched against the Tor network. They displayed the threats that are related the most to the concerned context. Additionally, Lagraa and François (2017) developed an approach that allows the discovery of port scanning behavior patterns and the classification of the features of port scans based on graph mining and modeling. The outcomes of their study were to provide security analysts with relevant data and information about the services that are jointly targeted and information concerning the relationship of the scanned ports. Such information was perceived as beneficial for evaluating the attacker's skills and strategy.

Niranjana et al. (2020) described the data formats for darknet traffic analysis, including basic and extended AGgregate and mode (AGM). In particular, they shed light on the 29-tuple numerical AGM data format, which efficiently analyzes the source I.P. address verified TCP connections. The analysis of the source I.P. validated TCP as a method in cybersecurity to identify the trends of the attack in the concerned network. Ozawa et al. (2020) shed light on the current composition of

the internet and the portion of the web held by the surface web, deep web, and dark web. They argued about how the dark web differs from the deep web. They shed light on the mechanism for accessing the deep web, tor browser, the dark web benefits, and some real-life applications.

Škrjanc et al. (2017) developed a method for cyber-attack large-scale monitoring using Cauchy possibility clustering. Seventeen (17) traffic features were extracted from the darknet packets, and the achieved detection rate was 98% for DDoS backscatter and 72.8% for non-DDoS backscatter communication through the use of support vector machines. Other studies on DDoS include (Cvitić et al. 2021; Mishra et al. 2021).

Balkanli et al. (2015) developed a classifier grounded upon decision trees to detect the backscatter DDoS events. CAIDA dataset was employed in this study to fulfil the training goals. Eight (8) out of twenty-one (21) features were extracted using symmetrical uncertainty and chi-square. Meanwhile, Ali et al. (2016) employed a neural network for detecting DDoS attacks using twenty features selected from the darknet traffic, utilizing NICT Japan. In another study, Furutani et al. (2014) employed eleven pieces of I.P. information /port features to detect DDoS backscatter communication; the authors used an SVM-based trained classifier. Their classifier achieved a 90% rate of accuracy.

Kumar et al. (2019) developed a framework that employs supervised machine learning and a concept drift detector. The classifiers could distinguish between benign and malignant traffic based on the experiment. Also, the classifiers could effectively detect capable and known threats and known, with a rate of accuracy of more than 99% accuracy. Association rule learning was utilized by Ozawa et al. (2020) to detect the regularities of those attacks from a large-scale darknet's massive stream data. They detected the behaviors of attacking hosts connected with well-known malware programs by examining the regularities in IoT-related indicators, such as the destination ports and types of services.

A deep neural network (DNN) typically requires much training data. It does not converge as fast as traditional machine learning algorithms (Young et al. 2018). The latter algorithms are deemed relatively simple to tune. Their output may offer interpretable results and lead to a better understanding of the problem, however, in terms of accuracy.

Darknet traffic identification and categorization using deep learning were proposed by Sarwar et al. (2021a) in their execution of data preprocessing on the complex, state-of-the-art dataset. Next, they examined various feature selection strategies to determine the most compelling features for detecting and categorizing darknet traffic. Then, they compared the performance of several finely tuned machine learning (ML) algorithms, such as decision tree, gradient boosting, random forest regressor (RFR), and extreme gradient boosting (XGB). A modified convolution-long short-term memory (CNN-LSTM) was then used. However, 89% of darknet traffic was classifiable using the CNN-LSTM, and XGB feature selection approaches. In grouping the high-dimensional data from network scanners, a deep representation learning approach was presented by Kallitsis et al. (2021). They used optimal classification trees for results interpretation, while the clustering results were used as "signatures" to detect structural darknet changes over time. The "signatures" were

assigned to the clustering results to detect structural changes in darknet activities, while an operational Network Telescope was used to test the proposed system's ability to identify high-impact cybersecurity incidents in the real world.

Rajawat et al. (2022) suggested a Dark Web Structural Patterns mining using neural networks and S^3VM for Criminal Network activity prediction, and the precision was 79%, respectively, and the percentage of dark web link prediction was 61%, which was still considered very low. Also, a high accuracy prediction was achieved using the random forest method (Abu Al-Haija et al. 2022).

Tor and VPN traffic was classified as the darknet, according to Habibi Lashkari et al. (2020), while all other types of internet traffic were classified as: "good traffic" (Cleartnet). To classify the dataset, they employed 61 features to create 8 by 8 grayscale images and then used a CNN. The CNN model was 0.94 accurate in determining whether traffic was coming from the darknet or was safe, and it was 86% accurate in determining what kind of application was causing the traffic. The application traffic was divided into categories: surfing, chatting, emailing, file transfer, P2P, audio streaming, video streaming, or voice over internet protocol (VOIP).

Meanwhile, Sarwar et al. (2021b) used a CNN with the long short-term memory (LSTM) and gated recurrent units (GRUs) deep learning techniques in an attempt to identify traffic and application type (GRU). On Tor, they used the synthetic minority oversampling technique (SMOTE) to address the issue of an imbalanced dataset. Twenty features were extracted using PCA, DT, XGB, and XGB+ before the data was fed into CNN-LSTM and CNN-GRU architectures, respectively. The input data was transformed into features using their CNN layer. To anticipate the following sequence, LSTM and GRU used these attributes as inputs. The best F1 scores were achieved with CNN-LSTM and XGB as the feature selector, with 0.96 traffic types and 0.89 application types accurately detected.

The CIC-Darknet2020 dataset's traffic patterns were the primary focus of the study (Iliadis and Kaifas 2021). They divided items into two or more groups using k-Nearest Neighbors (kNN), Multi-layer Perceptron (MLP), Radial Basis Function (RF), and Gradient Boosting (GB). They divided the gathered data into two categories—benign and darknet—so that binary categorization could be done. To address the issue of many classes, they chose to employ the first four traffic categories (Tor, non-Tor, VPN, or non-VPN). They discovered that RF was the most effective at classifying different traffic kinds, with F1 scores of 0.98 for binary and 98.61 for multi-class classification.

Demertzis et al. (2021) used weighted agnostic neural networks to classify the data (WANN). They did this by dividing the application category into 11 separate classes using the same dataset. They argued for less manual labor in the laborious process of creating artificial neural networks (ANN). WANN differs from conventional ANN in that its neurons' weights are constant. Instead, it makes incremental changes to the network's structure. WANN ranks the various architectures according to how well they function and how complex they are. The highest-ranked design is then used for the development of new network layers. Their WANN model had a classification accuracy of 0.92 percent for the application layer.

Sarkar et al. (2020) employed deep neural networks to distinguish between Tor traffic and other types of traffic using the UNB-CIC Tor and non-Tor dataset,

commonly known as ISCXTor2016 (Lashkari et al. 2017) (DNN). Two models were created. One (DNN-A) had three layers, whereas the other had five (DNN-B). DNN-B could distinguish between Tor samples and other samples 0.99 percent of the time, compared to DNN-98.81 A's percent accuracy. They used a four-layer DNN designed specifically for Tor samples to divide eight different types of applications into categories. The accuracy of this model was 0.95 of the time.

Another study Hu et al. (2020) used data from four different darknets. Their data set collected darknet traffic from eight applications (browsing, chat, email, file-transfer, P2P, audio, video, and VOIP) (Tor, I2P, ZeroNet, and Freenet). They used a three-tiered system to sort things. In the first layer, all traffic was divided into darknet traffic and other traffic. Second, samples from the darknet can be accurately identified, and lastly, the darknet itself can be identified in this layer. The third layer then groups the application categories for each darknet source. Some categorization methods are logistic regression (LR), random forest (RF), multi-layer perceptron (MLP), gradient boosting decision tree (GBDT), light gradient boosting (LightGB), XGA, and LSTM.

VPN and non-VPN traffic were examined in this study, and a classification system was developed using the new machine learning classifier techniques known as stacking ensembles learning. Machine learning techniques are used for VPN and Non-VPN classification.

Results from the experiment show that the study's classifiers can accurately distinguish between VPN and non-VPN traffic, (Almomani 2022). Previously discussed topics are summarised in Table 1.

The traditional machine learning algorithms could address the drawbacks of deep neural networks (DNN), especially concerning their poor performance. The same problem was also observed in algorithms in several domains related to Darknet traffic analysis and classification. Thus, the researcher conducted the present study to determine if the traditional algorithms could be employed to address the drawbacks of DNN and show a high-performance level compared to DNN. In this regard, a modern method called ensemble learning was proposed to show simplicity in setup. It has interpretable results and fast convergence on large and small datasets through deep learning for Darknet traffic analysis and classification. The details are provided below.

3 The proposed methodology

Ensemble and deep learning are currently popular methods for performing tasks like pattern recognition and predictive modeling. In this regard, deep learning is a powerful machine learning method that can extract lower-level features. It can feed such features to the following layer to identify the higher-level features that enhance the performance level. However, deep neural networks include drawbacks, including their various infinite architectures and hyper-parameters that could reduce the convergence speed on smaller datasets. The traditional machine learning algorithms can address such drawbacks but cannot show performance levels like those of deep neural networks. Ensemble methods shall be used to enhance the performance level

Table 1 Summary of related work

Research	Dataset	Method	Results
Demertzis et al. (2021)	CIC-Darknet2020	WANN	0.92 accuracy
Hu et al. (2020)	Self-generated	Binary classification using Hierarchical approach (LR, RF, MLP, GBDT, LightGB, XGB, LSTM and DT)	Layer 1: 0.99 accuracy Layer 2: 0.96 accuracy Layer 3: 0.92 accuracy
Iliadis and Kaifas (2021)	CIC-Darknet2020	Binary classification using kNN, MLP, RF, DT, and GB	F1-score: 0.98
Habibi Lashkari et al. (2020)	CIC-Darknet2020	Multiclass classification using CNN	Accuracy: 0.86
Sarkar et al. (2020)	ISCXTor2016	Multiclass using DNN	Accuracy: 0.956
Sarwar et al. (2021b)	CIC-Darknet2020	Multiclass: using CNN-LSTM CNN-GRU	F1-score 0.96 and .89 for internet traffic and application
Almomani (2022)	CIC-Darknet2020	techniques used for VPN and Non-VPN classification	The accuracy level is approximately 99%

for combining learners of multiple bases. Regarding super learning, it is an ensemble that finds the optimal mix of learning algorithms (Young et al. 2018). We thus proposed an approach of Darknet traffic analysis and classification system through ensemble learning adaptive techniques.

The proposed method involved the stacking approach, which is the most effective approach in dealing with the regression problem. Figure 6 accordingly displays the details of the proposed system. Next, the learning algorithms to be employed in the proposed system are identified. This work proposed can define a stacking ensemble scheme more formally in the following manner, in light of a set of N various learning algorithms $L_k, k = 1, \dots, N$ and the pair $\langle x, y \rangle$, with $x = (x_1, \dots, x_w)$ which represent the w recorded values and $y = (x_{w+1}, \dots, x_{w+h})$ the h values for prediction.

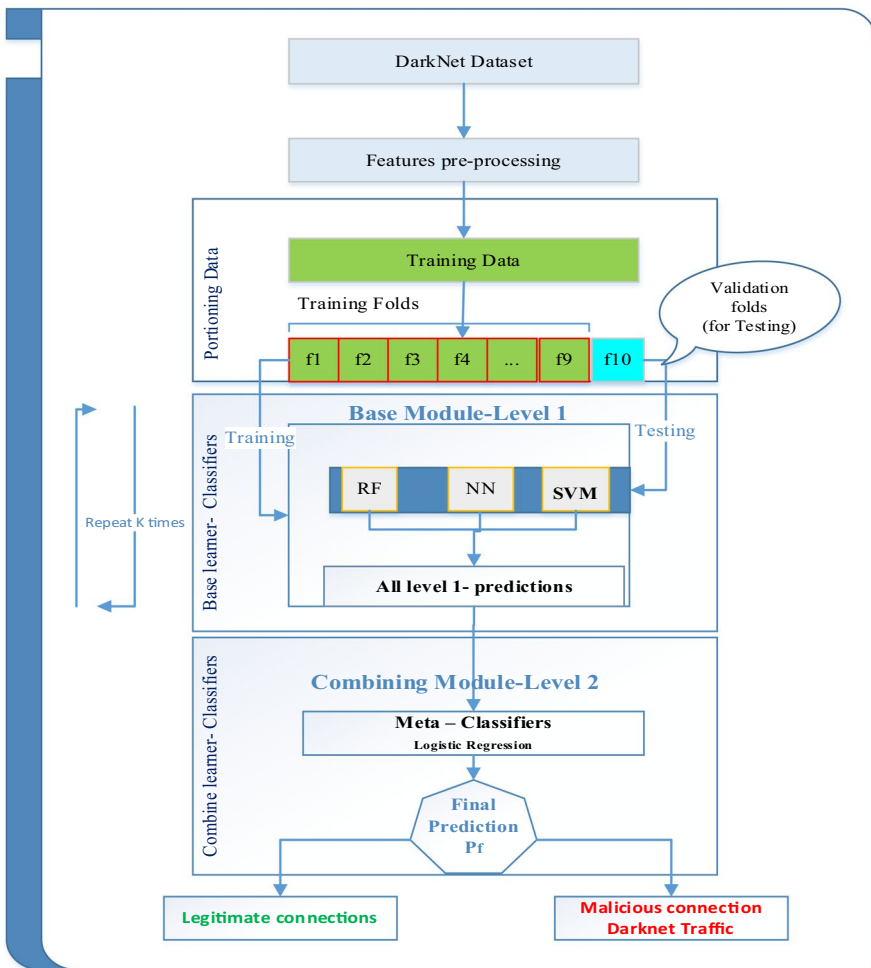


Fig. 6 Darknet traffic analysis and classification system based on stacking ensemble learning

Let m_{kj} , $k=1, \dots, N$, $j=1, \dots, h$ be the model induced through the learning algorithm L_k on x to predict x_{w+j} , and let f_j be the generalizer function that is responsible for having the models combined for having such value predicted. Then, f_j could be a generic function, like the model induced through a learning algorithm. Next, the estimated x_{w+j} value is provided through the expression: $\hat{x}_{w+j} = f_j(m_{1j}, \dots, m_{Nj})$. The full description of the darknet Traffic analysis and the system of classification based on the stacking ensemble learning can be viewed in Fig. 6.

Many contributions in this work are proposed, and as a result, the proposed system becomes stronger and more accurate as data grows. According to the experiment results, the classifiers can tell the difference between benign and malignant traffic with a high level of accuracy.

3.1 Darknet dataset

We have used the CIC-Darknet2020 dataset (Arash Habibi Lashkari and Abir Rahali 2020a, 2020b), to generate benign traffic and darknet ones. The darknet traffic comprises browsing, P2P, audio-stream, email, and chat. It involves VOIP, video-Stream, and transfer. The author combined the respective Tor and VPN traffic corresponding to darknet categories for the generated representative dataset. Namely, ISCXT or 2016 and ISCXVPN2016 have been amalgamated, and respective Tor and VPN traffic were combined in the corresponding categories of darknet. Table 1 displays the details of the categories of darknet traffic, precisely the details of the applications employed in producing the network traffic. There were 141,530 records as discussed below. Details related to the number of sampled benign traffic and darknet one can be viewed in Table 2 (Arash Habibi Lashkari and Abir Rahali 2020b).

3.2 Data and features pre-processing

3.2.1 Data cleaning phase

The full dataset comprising 77 features was employed in the study experiments. 61 features were used, as other features have many NAN, null, and zero-value records.

In this phase:

- We will be handling the missing values and replacing them with the mean value of the respective column.

Table 2 Darknet dataset

Dataset	# Of records after cleaning	Class
Non-Tor	93,356	0
NonVPN	23,863	1
VPN	22,919	2
Tor	1392	3

- Also, the null and NaN values record was deleted, and zero columns were removed.
- The columns with infinite (without value) were deleted as well.

3.3 The proposed ensemble learning system

This section's classification system for darknet traffic is based on stacking ensemble learning. Ensemble learning is more effective when there are differences between ensemble models, according to current empirical evidence. The stacked model's ensemble learning approach is the most frequently used stage of learning. This paper proposes a new two-tiered system with a base module and a combining module. These two types of modules were included in the system design.

3.3.1 Base module—level 1

The base module's job is represented in the proposed system by training, testing, and using a set for testing and training the base classifiers. Then, in level 2, the logistic regression algorithm will be used to make decisions. Detecting darknet traffic was the goal of this investigation. In this case, the issue stems from the binary classification of data. In this study, three base classifiers were used to create a base module for the proposed system. The random forest is one of three classifiers in this set (R.F.), artificial neural networks (ANNs) (Hopfield 1988), and support-vector networks (Cortes and Vapnik 1995). The binary classification problems can be solved by all of the algorithms.

3.3.1.1 Random forest (R.F.) Random forest (R.F.) was used for the first time (Breiman 2001). It refers to a group of decision trees that creates an ensemble of predictors. Thus, R.F. is mainly an ensemble of decision trees in which every tree is randomly trained separately on an independent training set. Every tree depends on the values of an input dataset sampled independently, with similar distribution for all trees. R.F. is a type of classification tree method, and the trees are shallow and constructed from several randomly selected samples. This method combines the results of these trees for predicting or classifying values, as shown in Fig. 7, while the pseudocode is displayed in Fig. 8. We have used the following parameter in the random forest (R.F.) algorithm.

*Random forest default parameters are: $n_estimators=100$, $criterion='gini'$, $min_samples_split=2$, $min_samples_leaf=1$, and $min_weight_fraction_leaf=0.0$. `class sklearn.ensemble.RandomForestClassifier (class_weight=None, ccp_alpha=0.0, max_samples=None, max_features='auto, max_depth=None, min_samples_split=2, oob_score=False, n_jobs=None, random_state=None, n_estimators=100, *, criterion='gini', max_leaf_nodes=None, min_impurity_decrease=0.0, min_samples_leaf=1, min_weight_fraction_leaf=0.0, min_impurity_split=None, bootstrap=True, verbose=0, warm_start=False,`*

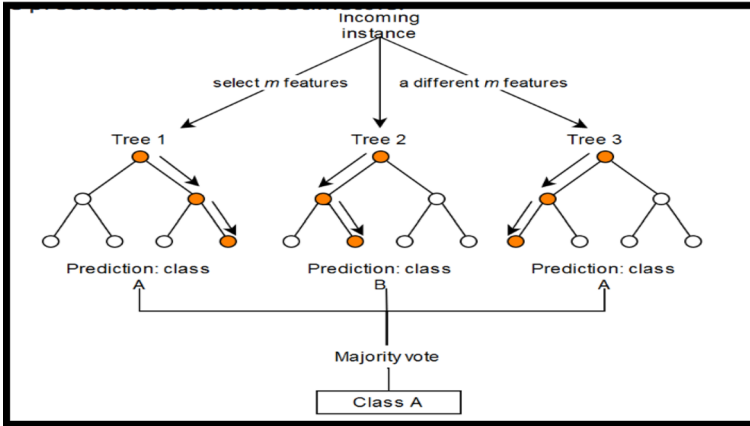


Fig. 7 Pseudocode for random tree algorithm in R.F. (Zhou 2019)

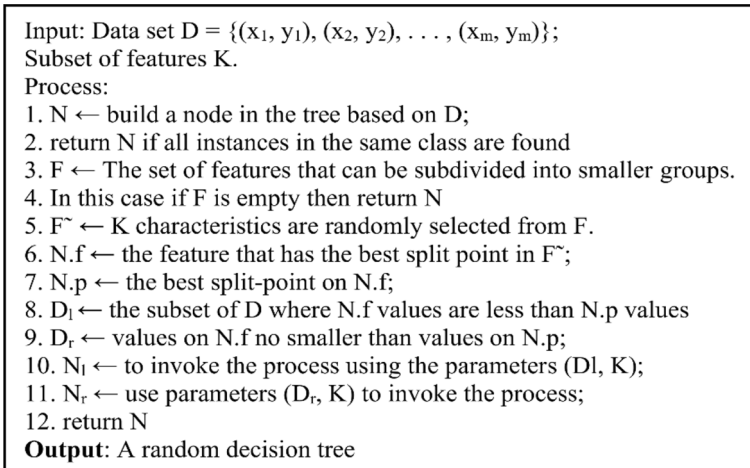


Fig. 8 Random forest inference for a simple classification example with $N_{tree}=3$ (Wood 2020)

3.3.1.2 Support-vector networks (SVM) SVM is a supervised learning model with associated learning algorithms that seek to analyze data for classification and perform regression analysis. SVM was designed by Vapnik et al. at AT&T Bell Laboratories (Cortes and Vapnik 1995). It is a prediction method that is considered the most robust. In the present study, the frameworks were used for statistical learning. In this regard, an SVM maps the training examples to points in space to maximize the width of the gap between the 2 categories. For performing the linear classification, SVMs can efficiently perform a non-linear classification by employing the kernel tricks. It maps their inputs into high-dimensional feature spaces. When having data that is unlabeled, supervised learning is not possible. In this case, it's required to have a learning approach that's unsupervised that seeks to find the natural clustering of the data into several groups. Then,

this approach seeks to map the new data into those groups. The clustering support-vector algorithm that Siegelmann and Vapnik developed implemented the statistics of support vectors generated in the support vector machines algorithm for categorizing the unlabeled data. It is a clustering algorithm commonly used in industrial-type applications (Ben-Hur et al. 2001), as shown in Fig. 9. The SVM parameters below were used.

SVM default parameters are: degree=3, gamma='auto', class sklearn.svm.SVC, C=1.0, kernel='rbf',(decision_function_shape="ovr", degree=3, gamma='auto_deprecated', coef0=0.0,random_state=None,C=1.0, kernel='rbf' shrinking=True,cache_size=200, class_weight=None probability=False, tol=0.001, verbose=False, max_iter=-1,)

Figure 9 shows the largest margin hyperplane in separating the instances of various classes. According to its definition: a margin is a minimal distance between examples of various classes and the classification hyperplane. Taken as an example: the hinge loss can be used to evaluate the fitness of a linear classifier.

Considering a linear classifier= y :

$$y = \text{sign}(w^T x + b), \text{ or abbreviated as } (w, b). \tag{1}$$

The hinge loss can be used to assess the data's fitness as follows:

$$\sum_{i=1}^m (\max\{0, 1 - y_i(w^T x_i + b)\}) \tag{2}$$

As an example, x_i instance Euclidean distance to $w^T x + b$ hyperplane is

$$\frac{|w^T x_i + b|}{\|w\|} \tag{3}$$

The polynomial kernel:

$$K(x_i, x_j) = x_i \cdot x_j^d \tag{4}$$

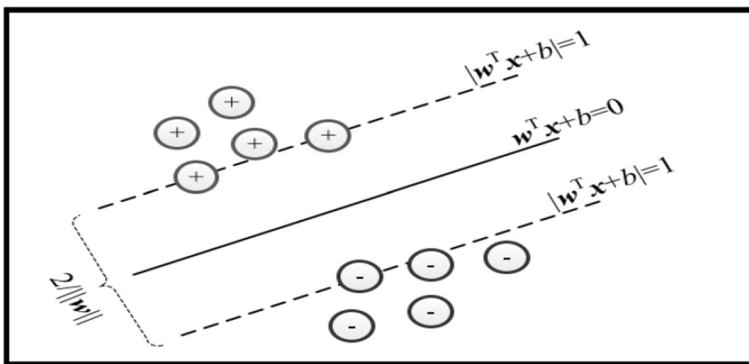


Fig. 9 Support-vector networks (SVM) (Zhou 2019)

where d is the degree of the polynomial, and the Gaussian kernel (or called RBF kernel):

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (5)$$

where σ is the parameter of the Gaussian width.

Kernel methods are the learning algorithms. SVMs are a kernel method that allows linear classifiers to be aided by a kernel trick.

3.3.1.3 Artificial neural networks (ANNs) included neural networks algorithm (Hopfield 1988) They are models of computational type, mimicking the functions and structure of the biological neural networks. Regarding the main computation unit, it is the neuron. It receives input from an external source or other nodes and seeks to compute an output. In output computation, the node shall apply a function called f (activation function) that introduces a non-linearity into the output. As shown in Fig. 10, the output shall be generated in case the inputs are above a specific threshold. We have used the parameter below in the Neural Networks algorithm as follows:

Default parameters are: ll_ratio=0.15, early_stopping=False, fit_intercept=True, max_iter=1000 alpha=0.0001, tol=0.001 and lass sklearn.linear_model.Perceptron (validation_fraction=0.1, n_iter_no_change=5 c.class_weight=None, warm_start=False, max_iter=1000, tol=0.001, shuffle=True,penalty=None, alpha=0.0001, verbose=0, eta0=1.0, n_jobs=None, random_state=0,ll_ratio=0.15, fit_intercept=True, early_stopping=False,)*

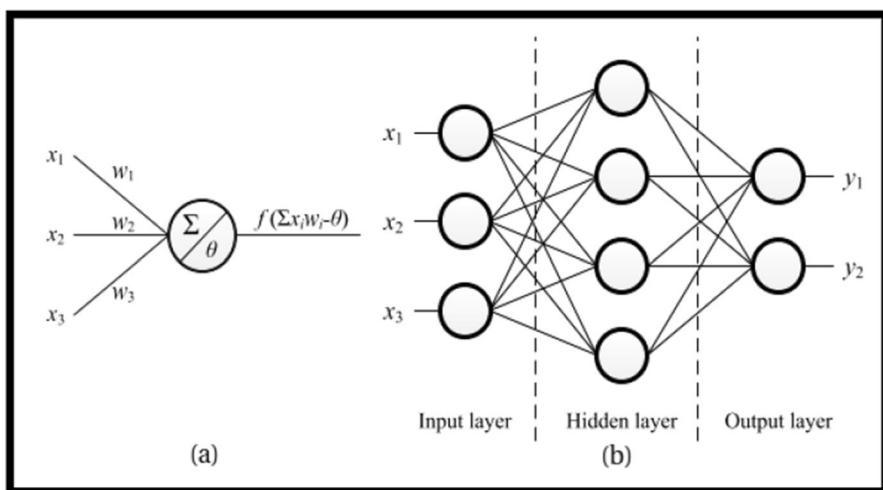


Fig. 10 a A neuron, and b a neural network (Zhou 2019)

It should be noted that both hidden neurons and output neurons are functional units, and the sigmoid function is a popular activation function for them, usually set as $f(x) = \frac{1}{1 + e^{-x}}$ (Zhou 2019).

$$f(x) = \frac{1}{1 + e^{-x}} \quad (6)$$

The prediction results of the base classifiers were employed by the ensemble learning stacked approach, as the input of the module called the combining module. Despite that, the researcher could not employ the whole dataset in testing and training the base classifiers and have the prediction results delivered to the aforementioned module for training. Regarding the potential model, it would have “seen” the test set. Therefore, there is a risk of over-fitting when the same data becomes the input for the prediction process, which affects the model validation significantly.

Regarding the generation methods of the stacking model, It is necessary to perform cross-validation. We chose the cross-validation K-fold method based on Fig. 6. It was first necessary for us to divide the initial dataset into two separate sets, the D_{test} , and the D_{train} . Using a K-fold cross-validation procedure, the D_{train} was divided into identical size K disjoint subsets. Folds are the names given to the subsets that make up a collection. It aims to keep the original dataset’s class scale intact. Every cross-validation was done on the D_{train} and the D_{test} for the training and testing phases, respectively. A single classifier $C_n(1, \dots, N)$ serves as an example of how many base classifiers can be used. One subset was used as a validation set D_{valid} , and the other subsets were used as training sets during the training phase. (K) Times the procedure was repeated. We form a matrix of prediction P_n ($n = 1, \dots, N$) for the entire prediction results on the validation sets. We used C_n to generate a classification matrix during the testing stage. We obtained (K) classification matrices and averaged them by rows to generate a matrix A_n ($n = 1, \dots, N$) after repeating this measure (K) times. The procedure was then applied to the remaining N classifiers in the same way as before. We will combine all of the prediction matrices, P_n , into a single training set. A modern test set (A) representing every level 1 classifier was generated by averaging all of the A_n . This prediction result represents all of the A_n . On to level 2 metaclassifier, it was passed from there. The number of folds (K) in the proposed method is 10, which reflects the actual dataset size and incorporates the knowledge of previous researchers. Representing the actual dataset size and joined with the experience of other researchers.

3.3.2 Combining module—level 2 by logistic regression (meta classifier)

The output of the ensemble will be fed into the meta-classifier level via stack generalization, allowing the logistic regression algorithm to determine the final decision. Regression analysis is based on the logistic regression algorithm. Logistic regression (Tolles and Meurer 2016): this model is used in statistics to estimate the likelihood of a given class or event, such as pass/fail or win/loss. Logistic regression is deemed a statistical model, which, in its primary form, delivers a logistic function for a dependent binary variable model. However, there

are numerous complex extensions. Through doing the analysis of regression, the logistic regression shall have the parameters of the estimated logistic model. From a mathematical perspective, the binary logistic model possesses a dependent variable with 2 possible values, namely the pass values and the fail values represented by an indicator variable. The two values are labeled 1 and 0. In the logistic model, the log-odds for the value labeled “1” is a linear combination of an independent variable or several ones (“predictors”).

Regarding the independent variable, it can serve as a binary variable (2 classes that are coded by an indicator variable) or as a continuous variable (any value that’s real). The corresponding probability of the value labeled 1 is within the range of 0–1. Thus, the function that converts the log odds into a probability is a logistic function for the labeling. The dependent variable in the binary logistic regression model has 2 levels (categorical). The outputs with more than 2 values are modeled by the logistic regression of multinomial nature if the multiple categories have been ordered.

Regarding the logistic regression model, it seeks to model the probability of the output in terms of input and does not perform a statistical classification. However, it may be employed as the classifier, for example, by selecting a cutoff value and categorizing the inputs with a probability higher than the cutoff as 1 class (Walker and Duncan 1967). However, combining the level strategies shall enhance the final classification capabilities, as shown in Fig. 11. The researcher employed the below parameter in Regression logistic algorithm as follows:

Logistic Regression (meta classifier) default parameters multi class = ‘auto’, max_iter = 100 and C = 1.0, random_state = 0, solver = ‘lbfgs’, meta_classifier = LogisticRegression(dual = False, fit_intercept = True, warm_start = False, intercept_scaling = 1, random_state = None, solver = ‘lbfgs’,

```

Input: Data set  $D = \{(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)\}$ ;
First-level learning algorithms  $L_1 \dots L_T$ ;
Second-level learning algorithm  $L$ .
Process:
1. for  $t = 1, \dots, T$ : % Apply the principles of training to a first-level learner.
2.    $h_t = L_t(D)$ ; % algorithm for first-level learning  $L_t$ 
3. end
4.  $Data' = \emptyset$ ; % Make a new data set collection.
5. for  $i = 1, \dots, n$ :
6.   for  $t = 1, \dots, T$ :
7.      $Z_{it} = h_t(X_i)$ ;
8.   end
9.    $Data' = Data' \cup ((Z_{i1}, \dots, Z_{iT}), Y_i)$ ;
10. end
11.  $h' = L(Data')$ ; % Applying  $h'$  to the second-level learner
    %  $L$  to the second-level learning algorithm
    % new data set  $Data'$ .
Output:  $H(X) = h'(h_1(X), \dots, h_T(X))$ 

```

Fig. 11 Binary classification with logistic regression (Hollemaans 2021)

$ll_ratio = None, C = 1.0, class_weight = None, max_iter = 100, multi_class = 'auto', n_jobs = None, penalty = 'l2', tol = 0.0001, verbose = 0$.

The sigmoid function is used to transform linear regression into L.R. An x-axis plots the value of x against the chance of each classification being correct. The Bernoulli distribution is assumed to be the distribution of y|x. L.R.'s formula is as follows:

$$f_x = \frac{1}{1 + e^{-\beta_0 + \beta_1 x}} \quad (7)$$

Here, $\beta_0 + \beta_1 x$ is similar to the linear model $y = ax + b$. The logistic function applies a sigmoid function to restrict the y value from a large scale to 0–1 (Sinnott et al. 2016).

Mixing independent base learners can significantly minimize the error rate. As an illustration, consider binary classification for the classes $-1, +1$. Assume that the ground-truth function f has an independent generalization error ϵ , i.e., for base classifier h_i ,

$$P(h_i(x) \neq f(x)) = \epsilon \quad (8)$$

That is, after combining the T number of such base classifiers according to:

$$H(x) = \text{sign} \left(\sum_{i=1}^T (h_i(x)) \right) \quad (9)$$

Only when at least half of the classifiers in ensemble H are incorrect will H make a mistake. As a result, the ensemble's generalization error is calculated using the Hoeffding inequality below:

$$P(H(x) \neq f(x)) = \sum_{k=0}^{T/2} \binom{T}{k} (1 - \epsilon)^k \epsilon^{T-k} \leq \exp c^2 \left(-\frac{1}{2} T 2\epsilon - 1 \right) 2 \quad (10)$$

If you look at Eq. 9, you can see that the generalization error decreases exponentially and ultimately approaches zero as T approaches infinity.

Finally, the proposed general Stacking procedure ensemble learning can be shown by pseudocode in Fig. 12.

4 Experiments results

The results of the dataset mentioned in part No. 3.1 are presented in this section, including the most noteworthy findings and the conclusion. Accuracy, precision, recall, AUC, F1 score and mean squared error, false negative rate (FNR), and False Positive rate (FP) were used to evaluate the performance of the basic methods and the ensemble scheme (FPR) [refer Table 3 and (Almomani 2018; Almomani et al. 2013)]. N-fold cross-validation (Chui et al. 2021) is used to assess the error rate of classifiers to determine the model performance. Meanwhile, tenfold

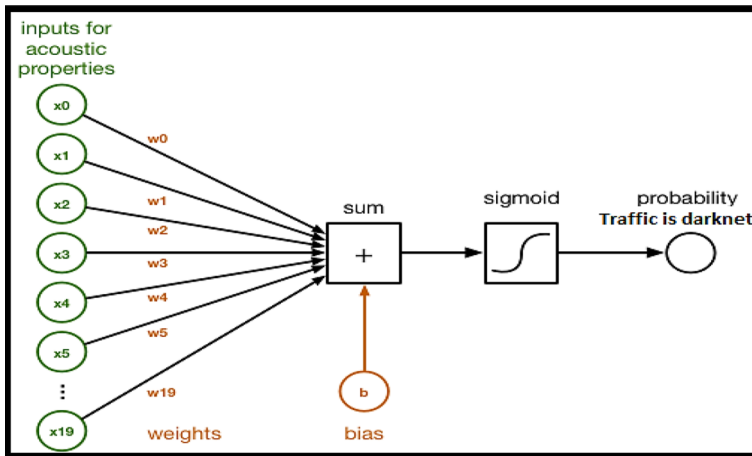


Fig. 12 Pseudocode of stacking ensemble used in our proposed system (Zhou 2019)

cross-validation involves randomly partitioning the dataset into N samples and performing assessments for N iterations. Each cycle selects $N-1$ samples for training, and the final sample is used to evaluate the classifier's accuracy.

Based on Tables 4, 5, and Fig. 13, the system proposed in this study has the best accuracy compared with other A.I. classifiers methods, at about 99%. In comparison, NN = 46%, LG = 81%, KNN = 95%, SVM = 80%, AND KNN = 92%. However, our proposed system shows results superior to other measurements, particularly precision and recall, as it reaches about 100%. Furthermore, the proposed method has the lowest error rate compared with other methods (MSE = 0.6%). The standard deviation in the training phase, which shows the stability of the proposed system, can be viewed in Table 6 and Fig. 14 below.

Table 7 and Fig. 14 show that when compared to other A.I. classification approaches, our proposed Ensemble System had the best standard deviation, which indicates that our proposed system was the most stable. A.I. classifiers with a lower accuracy rate than this one achieved a near-100% accuracy rate.

Tables 7, 8 and Fig. 15 clearly show that the suggested system has the best performance compared to existing A.I. classification methods—it reached around 96% accuracy while N.N. = 45%, LG = 81%, KNN = 92%, SVM = 79%, AND KNN = 92%. When compared to other methods, our system has the highest precision and recall rates, both of which are about 93%. In comparison to other methods, the proposed method has the lowest error rate, with an MSE of 3%. Table 9 and Fig. 16 indicate the standard deviation in the testing phase, which demonstrates the method's stability.

In comparison to other A.I. classifier approaches, our suggested Ensemble System has the best standard deviation, as shown in Table 9 and Fig. 16; it reached about 100% compared to other A.I. classifiers. For more details, we have done more experiments on FNR and FPR, as shown in Table 10.

Table 3 The Study's experiment measurements

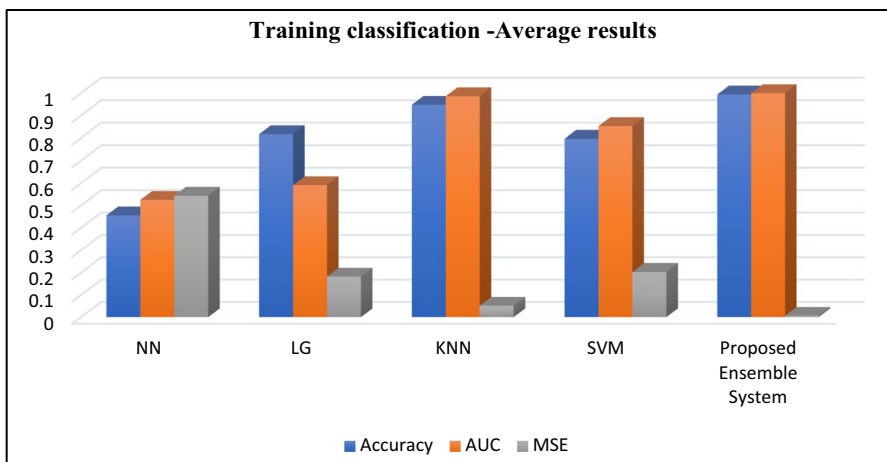
Rule number	Measurement	Equation	Meaning
1	Accuracy	$= \frac{ TP + TN }{ TP + TN + FP + FN }$	The percentage of correct predictions
2	Precision	$= \frac{ TP }{ TP + FP }$	The percentage of correct positive predictions
3	Recall	$= \frac{ TP }{ TP + FN }$	The percentage of positive labeled instances predicted as positive
4	AUC	TPR (true positive rate)/ recall/sensitivity	The ROC curve is plotted with TPR against the FPR
5	F1_Score	$= 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$	A measure of a test's accuracy calculated from the precision and test recall
6	Mean_squared_error (MSE) or root mean squared error (RMSE)	$\text{RMSE} = \sqrt{\sum_{i=1}^N \frac{(y_i - \hat{y}_i)^2}{N}}$	Where: n is the number of records input samples, y_i is the i_{th} actual output, \hat{y}_i is the i_{th} framework output, and RMSE=0 means that model output exactly matches the observed output. However: mean squared error (MSE) is used to evaluate the regression problem's accuracy
7	False negative rate (FNR)	$= \frac{FN}{P} = \frac{FN}{FN+TN}$	Condition positive (P) represents the data's number of real positive cases. Condition negative (N) represents the data's number of real negative cases. FNR, called the miss rate, is the test's probability that a true positive will be missed
8	False positive rate (FPR)	$= \frac{FP}{N} = \frac{FP}{FP+TN}$	Measures the proportion of correctly identified positives

Table 4 The classification results of various methods of machine learning—training data results

Method	Accuracy	AUC	MSE
NN	0.456 ± 0.0007	0.5251 ± 0.0009	−0.5437 ± 0.0007
LG	0.817 ± 0.0010	0.5909 ± 0.0017	−0.1829 ± 0.0010
KNN	0.948 ± 0.0003	0.9862 ± 0.0001	−0.0516 ± 0.0003
SVM	0.797 ± 0.0009	0.8529 ± 0.0052	−0.2031 ± 0.0009
Proposed ensemble system	0.994 ± 0.000601	0.9999 ± 0.000016	0.006 ± 0.000601

Table 5 Classification results of different machine learning Methods—average training data results

Method	Accuracy	AUC	MSE
NN	0.456	0.5251	0.5437
LG	0.817	0.5909	0.1829
KNN	0.948	0.9862	0.0516
SVM	0.797	0.8529	0.2031
Proposed ensemble system	0.994	0.9999	0.006

**Fig. 13** Based on the proposed ensemble system, the average outcomes of A.I. classifier approach against the proposed ensemble system

Based on Table 10 and Fig. 17, the class imbalance problem shall significantly impact the performance level of the conventional machine learning algorithms. For instance, the N.N. algorithm shows a TPR value of 0.585, and the K-neighbors classifier = 0.03. Thus, this means that many darknet traffic records samples (the class deemed minority) were misclassified as normal traffic records samples (the majority class). However, we can see that our proposed system has a low

Table 6 Results from various machine learning classification methods—training classification—standard deviation results

Method	Accuracy	AUC	MSE
NN	0.0007	0.0009	0.0007
LG	0.001	0.0017	0.001
KNN	0.0003	0.0001	0.0003
SVM	0.0009	0.0052	0.0009
Proposed ensemble system	0.0006	0.000016	0.000601

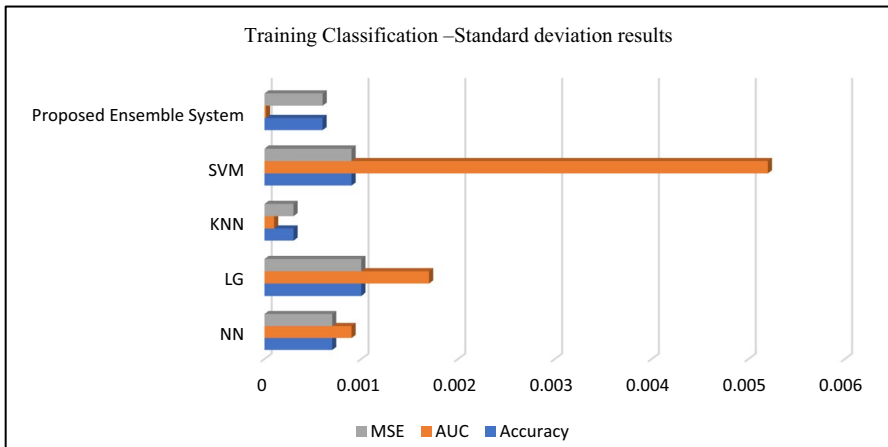


Fig. 14 Training classification—comparison of the proposed ensemble system with A.I. classifier standard deviation results

value of FPR = 0.1167 and low FNR = 0.0142 compared with other classifiers methods.

Wilcoxon signed-rank test (Rey and Neuhäuser 2011) was carried out to verify the accuracy of the models. This test evaluates the statistical significance of systematic pairwise differences among the models. In the case of the five susceptibility models, the *p*-value was utilized to evaluate the significance of differences among them to ensure that the results were statistically significant. In our investigation, the null hypothesis is the “there is no significant difference between classifiers” hypothesis. Therefore, the alternative hypothesis will be accepted rather than the null hypothesis if the value of *p* is greater than 0.05.

As shown in Table 11, there are statistically significant differences in the proposed ensemble system compared to the performance of the other selected machine learning model.

Table 7 Data analysis and classification results for a variety of machine learning approaches—testing data results

Method	Accuracy	Precision	Recall	AUC	F1_Score	MSE
NN	0.4563 ± 0.0063	0.1947 ± 0.0068	0.5298 ± 0.0080	0.5251 ± 0.0081	0.9295 ± 0.0040	0.5437 ± 0.0063
LG	0.8171 ± 0.0045	0.3685 ± 0.0413	0.5073 ± 0.0019	0.5909 ± 0.0065	0.8171 ± 0.0045	0.1829 ± 0.0045
KNN	0.9295 ± 0.0040	0.8437 ± 0.0127	0.8582 ± 0.0073	0.9345 ± 0.0044	0.9295 ± 0.0040	0.0705 ± 0.0040
RF	0.9684 ± 0.0021	0.9138 ± 0.0086	0.9456 ± 0.0047	0.9879 ± 0.0017	0.9684 ± 0.0021	0.0316 ± 0.0021
SVM	0.7969 ± 0.0033	0.8429 ± 0.0247	0.5136 ± 0.0009	0.8486 ± 0.0060	0.7969 ± 0.0033	-0.2031 ± 0.0033
Proposed ensemble system	0.9674 ± 0.0019	0.9316 ± 0.0074	0.9348 ± 0.0032	0.9845 ± 0.0019	0.9674 ± 0.0020	0.0325 ± 0.0019

Table 8 Classification results of different machine learning methods—average testing data results

Method	Accuracy	Precision	Recall	AUC	F1_Score	MSE
NN	0.4563	0.1947	0.5298	0.5251	0.9295	0.5437
LG	0.8171	0.3685	0.5073	0.5909	0.8171	0.1829
KNN	0.9295	0.8437	0.8582	0.9345	0.9295	0.0705
SVM	0.7969	0.8429	0.5136	0.8486	0.7969	0.203
Proposed ensemble system	0.9674	0.9316	0.9348	0.9845	0.9674	0.0325

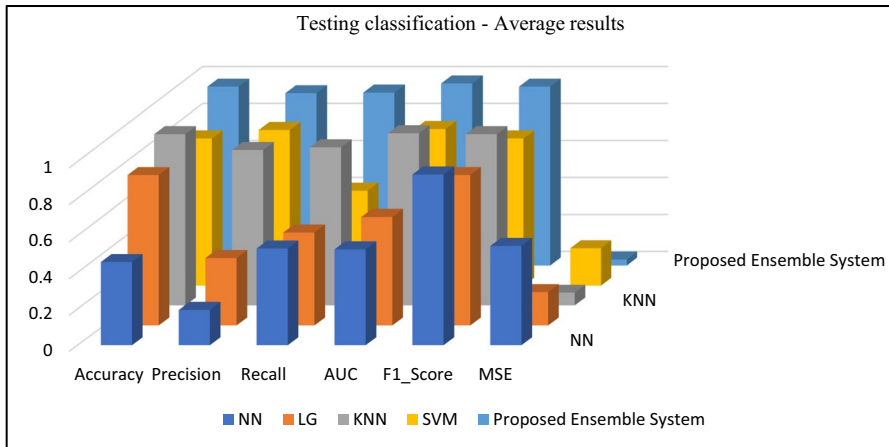


Fig. 15 Testing classification—compared to the proposed Ensemble system, the average results of A.I. classifiers methods

Table 9 Standard deviation during the testing phase of classification results from various machine learning methods

Method	Accuracy	Precision	Recall	AUC	F1_Score	MSE
NN	0.0063	0.0068	0.008	0.0081	0.004	0.0063
LG	0.0045	0.0413	0.0019	0.0065	0.0045	0.0045
KNN	0.0040	0.0127	0.0073	0.0044	0.004	0.004
SVM	0.0033	0.0247	0.0009	0.006	0.0033	0.0033
Proposed ensemble system	0.0019	0.0074	0.0032	0.0019	0.002	0.0019

5 Conclusion

Darknet is a portion of allocated I.P. routed space at which none of the servers or active services resides. Stacking ensemble learning, a new machine learning classifier technique, was proposed in this paper as a method for analyzing darknet traffic and classifying attackers. In ensemble learning, multiple learning mechanisms can

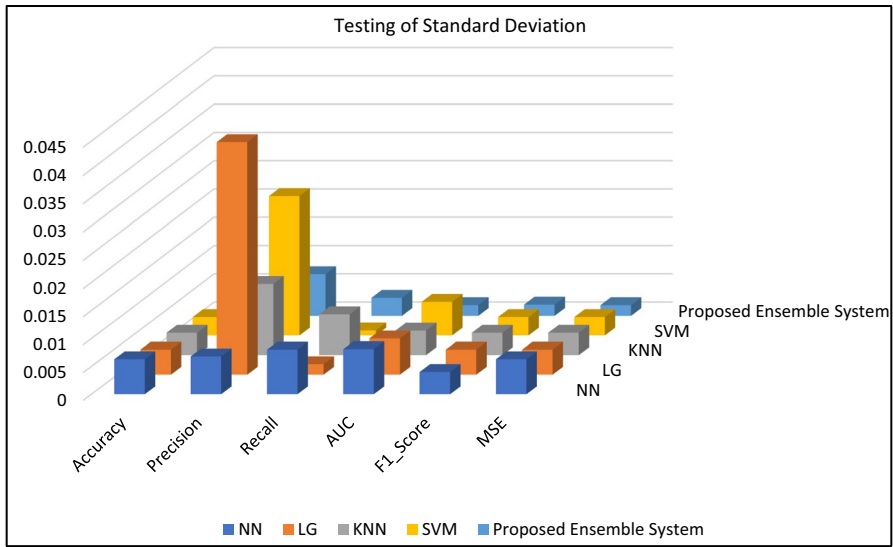


Fig. 16 Testing classification—comparison of A.I. classifier standard deviations with that of the proposed Ensemble system

Table 10 Comparison of false positive and false negative rates (misclassification)—testing data results

Method	FNR	FPR
NN	0.585063	0.355294
LG	0.008727	0.976657
KNN	0.030345	0.253347
SVM	0.0014	0.9713
Proposed ensemble system	0.0142	0.1167

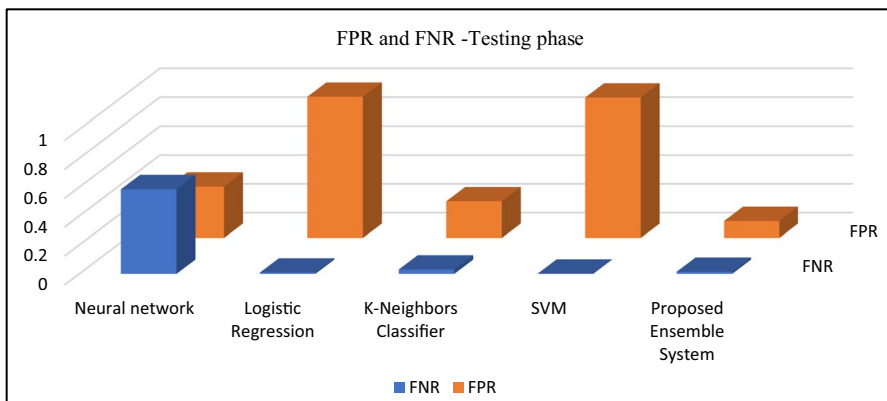


Fig. 17 Testing classification—FPR and FNR results based on A.I. classifiers methods compared with proposed ensemble system

Table 11 Comparing the proposed ensemble's performances with other machine learning models using the Wilcoxon signed-rank test

Comparison	Significance
NN versus proposed ensemble system	Yes
LG versus proposed ensemble system	Yes
KNN versus proposed ensemble system	Yes
SVM versus proposed ensemble system	Yes

be combined to produce more predictions that are accurate. This study utilized a two-tiered learning stacking scheme that included both individual and group learning. The 1st-level technique's prediction was passed on to a higher-level technique, which combined it with other predictions to produce a final prediction. Based on meta-classifiers, such as logistic regression, the top-level technique used three base learning methods, such as neural networks, random forests, and support vector machines, to generate the final predictions. Precision, accuracy, and root-mean-square error (RMSE) were the primary metrics used to arrive at a prediction. Comparisons were made between the outcomes of the ensemble and the single technique.

In comparison to other techniques, results from the ensemble scheme were more impressive. The researcher observed that the ensemble method could handle small historic windows, which means that the proposed system becomes stronger and more accurate as data size increases. The experiment results clearly showed that the classifiers could distinguish between benign and malignant traffic. Specifically, during the training phase, the classifiers could effectively detect unknown and known threats with more than 99% accuracy, and during the testing phase, the achieved accuracy rate of the classifiers was 96%. To this end, the present study proposed investigating other ensemble schemes by employing various other methods in future studies, for instance, using methods based on S.P. Theory of Intelligence and support vector machines against certain types of attacks.

We have some limitations because of performance and privacy concerns; we are unable to do deep packet inspection in our research. Using a variety of methodologies, including those based on the S.P. Theory of Intelligence and support vector machines, we plan to examine other ensemble schemes in the near future. The proposed solution will be evaluated using various databases and different types of attacks to see if it is effective.

Acknowledgements Al-Balqa Applied University in Jordan supported the research reported in this publication. Grant #: **DSR-2021#398**.

References

- Abu Al-Haija Q, Krichen M, Abu Elhaija W (2022) Machine-learning-based darknet traffic detection system for IoT applications. *Electronics* 11(4):556
- Ali SHA, Ozawa S, Ban T, Nakazato J, Shimamura J (2016) A neural network model for detecting DDoS attacks using darknet traffic features. In: 2016 International joint conference on neural networks (IJCNN).

- Alieyan K, Anbar M, Almomani A, Abdullah R, Alauthman M (2018) Botnets detecting attack based on DNS features. In: 2018 International Arab conference on information technology (ACIT).
- Al-Kasassbeh M, Mohammed S, Alauthman M, Almomani A (2020) Feature selection using a machine learning to classify a malware. In: Gupta BB, Perez GM, Agrawal DP, Gupta D (eds) Handbook of computer networks and cyber security. Springer, Berlin, pp 889–904
- Almomani A (2018) Fast-flux hunter: a system for filtering online fast-flux botnet. *Neural Comput Appl* 29(7):483–493
- Almomani A (2022) Classification of virtual private networks encrypted traffic using ensemble learning algorithms. *Egypt Inf J* 23:57
- Almomani A, Gupta BB, Atawneh S, Meulenber A, Almomani E (2013) A survey of phishing email filtering techniques. *IEEE Comm Surv Tutor* 15(4):2070–2090
- Al-Nawasrah A, Al-Momani A, Meziane F, Alauthman M (2018) Fast flux botnet detection framework using adaptive dynamic evolving spiking neural network algorithm. In: 2018 9th international conference on information and communication systems (ICICS).
- Arash Habibi Lashkari GK, Abir Rahali (2020a) CIC-Darknet2020a. In: Canadian institute for cybersecurity. Retrieved July 1 2021 from <https://www.unb.ca/cic/datasets/darknet2020a.html>
- Arash Habibi Lashkari GK, Abir Rahali (2020b) DIDarknet: a contemporary approach to detect and characterize the darknet traffic using deep image learning. In: 10th international conference on communication and network security, Tokyo, Japan. <https://www.unb.ca/cic/datasets/darknet2020b.html>
- Ardabili S, Mosavi A, Várkonyi-Kóczy AR (2019) Advances in machine learning modeling reviewing hybrid and ensemble methods. In: International conference on global research and education.
- Balkanli E, Zincir-Heywood AN, Heywood MI (2015) Feature selection for robust backscatter DDoS detection. In: 2015 IEEE 40th local computer networks conference workshops (LCN workshops).
- Ben-Hur A, Horn D, Siegelmann HT, Vapnik V (2001) Support vector clustering. *J Mach Learn Res* 2(12):125–137
- Bou-Harb E, Assi C, Debbabi M (2016) Csc-detector: a system to infer large-scale probing campaigns. *IEEE Trans Dependable Secur Comput* 15(3):364–377
- Bou-Harb E, Husák M, Debbabi M, Assi C (2017) Big data sanitization and cyber situational awareness: a network telescope perspective. *IEEE Trans Big Data* 5:439
- Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
- Cambiaso E, Vaccari I, Patti L, Aiello M (2019) Darknet security: a categorization of attacks to the tor network. In: ITASEC.
- Chui KT, Gupta BB, Vasant P (2021) A genetic algorithm optimized rnn-lstm model for remaining useful life prediction of turbofan engine. *Electronics* 10(3):285
- Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20(3):273–297
- Cvitić I, Peraković D, Gupta B, Choo K-KR (2021) Boosting-based DDoS detection in internet of things systems. *IEEE Internet Things J* 9:2109
- Dainotti A, King A, Claffy K, Papale F, Pescapé A (2014) Analysis of a “0” stealth scan from a botnet. *IEEE/ACM Trans Networking* 23(2):341–354
- Demertzis K, Tsiknas K, Takezis D, Skianis C, Iliadis LJE (2021) 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* 10(7):781
- Dietterich TG (2000) Ensemble methods in machine learning. In: International workshop on multiple classifier systems.
- Divina F, Gilson A, Gómez-Vela F, García Torres M, Torres JF (2018) Stacking ensemble learning for short-term electricity consumption forecasting. *Energies* 11(4):949
- Du P, Xia J, Zhang W, Tan K, Liu Y, Liu S (2012) Multiple classifier system for remote sensing image classification: a review. *Sensors* 12(4):4764–4792
- Furutani N, Ban T, Nakazato J, Shimamura J, Kitazono J, Ozawa S (2014) Detection of DDoS backscatter based on traffic features of darknet TCP packets. In: 2014 Ninth Asia Joint conference on information security.
- Gdata (2022) What actually is the Darknet? In: Gdata-rust in gurman sicherheit. Retrieved June 7 2022 from <https://www.gdatasoftware.com/guidebook/what-is-the-darknet-exactly>
- Habibi Lashkari A, Kaur G, Rahali A (2020) DIDarknet: a contemporary approach to detect and characterize the darknet traffic using deep image learning. In: 2020 the 10th international conference on communication and network security.
- Hansen LK, Salamon P (1990) Neural network ensembles. *IEEE Trans Pattern Anal Mach Intell* 12(10):993–1001

- Hollemaans M (2021) Binary classification with logistic regression. In: Tensorflow. Retrieved 1–3 from <http://machinethink.net/blog/tensorflow-on-ios/>
- Hopfield JJ (1988) Artificial neural networks. *IEEE Circuits Devices Mag* 4(5):3–10
- Hu Y, Zou F, Li L, Yi P (2020) Traffic classification of user behaviors in tor, i2p, zeronet, freenet. In: 2020 IEEE 19th international conference on trust, security and privacy in computing and communications (TrustCom).
- Iliadis LA, Kaifas T (2021) Darknet traffic classification using machine learning techniques. In: 2021 10th international conference on modern circuits and systems technologies (MOCAST).
- Kallitsis M, Honavar V, Prajapati R, Wu D, Yen J (2021) Zooming into the darknet: characterizing internet background radiation and its structural changes. <https://arxiv.org/abs/2108.00079>
- Kumar S, Vranken H, van Dijk J, Hamalainen T (2019) Deep in the dark: a novel threat detection system using darknet traffic. In: 2019 IEEE International conference on big data (big data).
- Lagraa S, François J (2017) Knowledge discovery of port scans from darknet. In: 2017 IFIP/IEEE symposium on integrated network and service management (IM).
- Lashkari AH, Draper-Gil G, Mamun MSI, Ghorbani AA (2017) Characterization of tor traffic using time based features. In: ICISSp
- Mishra A, Gupta N, Gupta B (2021) Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX controller. *Telecommun Syst* 77(1):47–62
- Morgan S (2021) Cybercrime to cost the world \$10.5 trillion annually by 2025. *Cybercrime Magazine*. Retrieved August 26 2021 from <https://cybersecurityventures.com/cybercrime-damage-costs-10-trillion-by-2025>
- Niranjana R, Kumar VA, Sheen S (2020) Darknet traffic analysis and classification using numerical AGM and mean shift clustering algorithm. *SN Comput Sci* 1(1):16
- oreilly (2022) Bagging—building an ensemble of classifiers from bootstrap samples. orilly. Retrieved June 6 2022 from <https://www.oreilly.com/library/view/python-real-world-data/9781786465160/ch41s04.html>
- Ozawa S, Ban T, Hashimoto N, Nakazato J, Shimamura J (2020) A study of IoT malware activities using association rule learning for darknet sensor data. *Int J Inf Secur* 19(1):83–92
- Pang R, Yegneswaran V, Barford P, Paxson V, Peterson L (2004) Characteristics of internet background radiation. In: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement.
- Patel A (2020). <https://medium.com/ml-research-lab/stacking-ensemble-meta-algorithms-for-improve-predictions-f4b4cf3b9237>. ML Research Lab. Retrieved June 7 2022 from <https://medium.com/ml-research-lab/stacking-ensemble-meta-algorithms-for-improve-predictions-f4b4cf3b9237>
- Perrone MP, Cooper LN (1992) When networks disagree: Ensemble methods for hybrid neural networks. World scientific, Hackensack
- Ponti Jr MP (2011) Combining classifiers: from the creation of ensembles to the decision fusion. In: 2011 24th SIBGRAPI conference on graphics, patterns, and images tutorials.
- Rajawat AS, Bedi P, Goyal S, Kautish S, Xihua Z, Aljuaid H, Mohamed AW (2022) Dark web data classification using neural network. *Comput Intell Neurosci* 2022:1–11
- Rey D, Neuhäuser M (2011) Wilcoxon-signed-rank test. In: Lovric M (ed) *International encyclopedia of statistical science*. Springer, Berlin, pp 1658–1659
- Sahoo SR, Gupta BB (2021) Multiple features based approach for automatic fake news detection on social networks using deep learning. *Appl Soft Comput* 100:106983
- Sarkar D, Vinod P, Yerima SY (2020) Detection of Tor traffic using deep learning. In: 2020 IEEE/ACS 17th international conference on computer systems and applications (AICCSA).
- Sarwar MB, Hanif MK, Talib R, Younas M, Sarwar MU (2021a) DarkDetect: Darknet traffic detection and categorization using modified convolution-long short-term memory. *IEEE Access* 9:113705–113713
- Sinnott R, Duan H, Sun Y (2016) Chapter 15-a case study in big data analytics: exploring twitter sentiment analysis and the weather. *Big Data*, 357–388
- Škrjanc I, Ozawa S, Dovžan D, Tao B, Nakazato J, Shimamura J (2017) Evolving cauchy possibilistic clustering and its application to large-scale cyberattack monitoring. In: 2017 IEEE symposium series on computational intelligence (SSCI).
- Tolles J, Meurer WJ (2016) Logistic regression: relating patient characteristics to outcomes. *JAMA* 316(5):533–534
- UNODC (2021) Darknet cybercrime threats to Southeast Asia 2020 (UNODC report: darknet cybercrime is on the rise in Southeast Asia, Issue). <https://www.unodc.org/southeastasiaandpacific/en/2021/02/darknet-cybercrime-southeast-asia/story.html>

- Walker SH, Duncan DB (1967) Estimation of the probability of an event as a function of several independent variables. *Biometrika* 54(1–2):167–179
- Wang Q, Chen Z, Chen C (2011) Darknet-based inference of internet worm temporal characteristics. *IEEE Trans Inf Forensics Secur* 6(4):1382–1393
- Wood T (2020) Random forests. *Deep AI*. Retrieved 1–3 from <https://deeptai.org/machine-learning-glossary-and-terms/random-forest>
- Woźniak M, Grana M, Corchado E (2014) A survey of multiple classifier systems as hybrid systems. *Inf Fusion* 16:3–17
- Young S, Abdou T, Bener A (2018) Deep super learner: a deep ensemble for classification problems. In: *Canadian conference on artificial intelligence*.
- Zhang R, Yang C, Pang S, Sarrafzadeh H (2017) Unitecdeamp: flow feature profiling for malicious events identification in darknet space. In: *International conference on applications and techniques in information security*.
- Zhang Z (2019) Boosting algorithms explained, theory, implementation, and visualization. Retrieved June 7 2022 from <https://towardsdatascience.com/boosting-algorithms-explained-d38f56ef3f30>
- Zhou Z-H (2019) *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, Boca Raton

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.