Yalvaç Akademi Dergisi

https://dergipark.org.tr/tr/pub/yalvac

COMPARISON OF MACHINE LEARNING ALGORITHMS FOR DETECTION OF DATA EXFILTRATION OVER DNS

Enes AÇIKGÖZOĞLU[1]

[1] Computer Programming, Keçiborlu Vocational School, Isparta University of Applied Sciences, Isparta, TURKIYE

DOI: 10.57120/yalvac.1507402

ÖZET

Günümüzde bilgisayarlar, iş süreçlerinde ve ev kullanıcıları için vazgeçilmezdir. İnternetin yaygın kullanımı, eğitimden araştırmaya pek çok alanda kolaylık sağlamaktadır. Ancak, kullanıcıların çoğu teknik güvenlik önlemlerinden habersizdir ve interneti bilinçsizce kullanmaktadır. Bu durum, siber saldırılara karşı yetersiz güvenlik önlemlerine yol açmaktadır. Bilinçli ve güvenli internet kullanımı için çeşitli eğitimler düzenlenmekte, ancak bu çabalar yeterli olmamaktadır. Bu nedenle, siber olayları tespit edebilecek ve güvenlik açıklarını kapatacak yapay zeka temelli çözümler gerekli hale gelmektedir. DNS tünelleme, zararlı yazılımların internet üzerinden veri sızdırmak için kullandığı bir yöntemdir. Zafiyetli bilgisayarlar, yanlış DNS sunucularından IP adresi öğrenerek kullanıcıları zor durumlara düşürebilmektedir. Bu tünellemeyi tespit etmek için yenilikçi yöntemler geliştirilmiştir. Bazı yöntemler DNS üzerinden düşük ve yavaş veri sızıntısını gerçek zamanlı tespit edebilmektedirler. Ayrıca, paket uzunluğu ve belirli özellikleri kullanarak yüksek doğruluk ve F-skoru elde eden hibrit DNS tünelleme tespit sistemleri bulunmaktadır. Önbellek özelliklerine duyarlı özelliklere dayalı yöntemler ise düşük yanlış tespit oranlarıyla DNS tünelleme trafiğini etkili bir şekilde karakterize eder. Bu yöntemler, internet güvenliği konusunda etkili stratejiler sunmaktadır. Yapılan bu çalışmada CIC-Bell-DNS-EXF-2021 veri seti üzerinde makine öğrenimi algoritmalarının DNS tünelleme ataklarını tespit etme durumları araştırılmıştır.

Anahtar Kelimeler- CIC-Bell-DNS-EXF-2021, Makine Öğrenmesi, DNS Tünelleme

ABSTRACT

Nowadays, computers are indispensable for business processes and home users. The widespread use of the Internet provides convenience in many areas from education to research. However, most of the users are unaware of technical security measures and use the Internet unconsciously. This situation leads to inadequate security measures against cyber-attacks. Various trainings are organised for conscious and safe internet use, but these efforts are not enough. Therefore, artificial intelligence-based solutions that can detect cyber incidents and close security gaps are becoming necessary. DNS tunnelling is a method used by malware to leak data over the internet. Vulnerable computers can put users in difficult situations by learning IP addresses from the wrong DNS servers. Innovative methods have been developed to detect this tunnelling. Some methods can detect low and slow data leakage through DNS in real time. There are also hybrid DNS tunnelling detection systems that achieve high accuracy and F-score using packet length and specific features. Feature-based methods sensitive to cache characteristics effectively characterise DNS tunnelling traffic with low false detection rates. These methods offer effective strategies for internet security. In this study, the detection of DNS tunnelling attacks by machine learning algorithms on the CIC-Bell-DNS-EXF-2021 dataset was investigated.

Keywords- CIC-Bell-DNS-EXF-2021, Machine Learning, DNS Tunnelling

1. INTRODUCTION

Cyber attacks have become increasingly sophisticated with the widespread use of the Internet today. In this context, DNS (Domain Name System) tunnelling is one of the methods that pose a major threat to cyber security. DNS tunnelling is a technique used by malicious software for data exfiltration and command and control operations. The fact that DNS is one of the cornerstones of the Internet makes such attacks even more dangerous. Therefore, detecting and preventing DNS tunnelling attacks has become an important part of cyber security strategies.

Detecting DNS tunnelling attacks can be quite challenging with traditional security methods. Since such attacks are usually carried out with low and slow data transfer, they are difficult to detect with ordinary network traffic analyses. In addition, because DNS tunnelling is very similar to standard DNS traffic, firewalls and other security measures often miss these attacks. This suggests the need for more innovative and advanced techniques for early detection and prevention of DNS tunnelling attacks.

Users who do not use the Internet consciously and safely can often encounter bad situations and are negatively affected mentally and socially. Users who surf the internet unconsciously infect their computers with viruses, trojans, etc. They become vulnerable by infecting them with bad software. Vulnerable computers can perform many operations in the background without the user's knowledge. In order to surf the Internet and access a web page, it is necessary to know the IP address of the relevant server. DNS service allows users to navigate on the web using domain names instead of memorising IP addresses. Vulnerable user computers can put users in difficult situations by learning the required IP address from the wrong DNS servers when they want to browse the internet or access a file.

Several research papers propose innovative methods to detect low and slow data leakage and tunnelling through DNS. One approach is the Information Based Heavy Hitters method, which offers real-time detection based on live estimates of the information transmitted to registered domains. This method can detect infiltration rates as slow as 0.7B/s with minimal false positives [1]. Furthermore, a hybrid DNS tunnelling detection system has been developed that uses packet length and selected features for network traffic analysis, achieving high accuracy and F-score in detecting DNS tunnelling attacks [2]. Furthermore, a cache feature-aware feature-based DNS tunnelling detection method is introduced, which demonstrates effective characterisation of DNS tunnelling traffic and offers rule-based and LSTM-based filters for detection with low false detection rates [3]. Collectively, these methods provide effective strategies to combat low and slow data leakage and tunnelling through DNS.

DNS tunnelling can be effectively detected through various methodologies, including the use of cloud-based resources for monitoring and anomaly identification [4] [5]. One approach is to leverage unsupervised machine learning models to detect anomalies in DNS traffic, enabling detection of malicious activities such as data leakage and command and control channels [5]. Furthermore, monitoring the current lengths of DNS subdomains can serve as an effective countermeasure against DNS tunnelling, especially for general Internet users. This can be achieved by analysing the distribution patterns of subdomain lengths to distinguish normal traffic from tunnelling activities [3]. By combining these detection techniques, organisations can build robust monitoring systems to effectively detect and prevent DNS tunnelling and improve their overall cybersecurity posture.

Indicators of DNS tunnelling activities include data exfiltration, misuse of DNS protocols for command and control operations, and bypassing security measures such as captive portals [6]. Detection techniques usually involve monitoring network flows, analysing flow-derived variables and using statistical methods for anomaly detection [7]. With tools such as Elasticsearch facilitating the detection of DNS tunnelling, the use of unique hostnames can be used as a compromise indicator. These indicators then enable blacklists to be applied by administrators who are made aware of the incident [8]. Machine learning frameworks such as neural networks have been proposed for the construction of DNS tunnelling detectors that can effectively distinguish tunnelling DNS packets from normal packets with high accuracy without generating false positives. Furthermore, scoring DNS nameservers based on query data over time can help identify potential DNS tunnelling activities for further investigation and mitigation [9].

In order to detect DNS tunnelling attacks and to test different machine learning algorithms in this area, the CIC-Bell-DNS-EXF-2021 dataset was created by academics at the University of New Brunswick, Canada. The dataset consists of 641642 benign samples, 53978 mild attack samples and 323698 severe attack samples. Random Forest (RF), Gaussian Naive Bayes (GNB), Multilayer Perceptron (MLP), Logistic Regression (LR) and Support Vector Machine (SVM) machine learning algorithms were experimentally analysed on the dataset [10].

In a study running on devices with Android operating system and proposing an isolated forest-based DNS tunnel detection method, DNS tunnel traffic was collected and analysed. Depending on the DNS request and DNS response, features were extracted and a feature set was created. In this study, KRTunnel, a DNS tunnel detector for mobile devices with android operating system, is proposed. With the proposed KRTunnel, DNS tunnel traffic can be identified with %98.1 accuracy [11].

Mitsuhashi et al. developed a malicious DNS tunnelling tool recognition system by analysing DNS traffic over HTTPS. The developed system is based on hierarchical machine learning classification and XGBoost, LightGBM and CatBoost machine learning models with gradient boosting decision tree (GBDT) algorithm, which are known for their high accuracy and short training time, are used for classification. According to the evaluation results, the DNS tunnelling tool system identified malicious DNS traffic with a classification accuracy of %98.02 [12].

In order to intelligently detect DNS tunnel attacks, a proposed system analyses domain features as load-based features, traffic-based features and resolution-based features and extracts representative features from DNS

servers. To classify the DNS tunnel, the Random Forest algorithm is mainly used and a fusion RF-LR model is proposed by taking logistic regression process for each decision tree nodes. The proposed RF-LR model is said to have high detection accuracy and the classification accuracy is given as %98.81 [13].

In a study conducted in 2023, a hybrid DNS tunnelling detection system was proposed with features selected for network traffic and packet length. Test bed and Tabu-PIO feature selection algorithms are used in the system. The proposed approach was shown to achieve %98.3 accuracy and %97.6 F-score on DNS tunnelling datasets [14].

A two-step system using machine learning algorithms and basic features to detect hidden DNS tunnelling traffic over HTTPS was proposed by Wang et al. In the first step, the characteristics of DOH traffic are identified and machine learning methods are used. In the second step, self-collected data and public datasets are used to validate the system. They showed that the proposed system achieved up to %99 accuracy and recall rate [15].

Lal et al. proposed a new DNS tunnel detection technique using the integration of Convolutional Neural Networks (CNN) and Support Vector Machines (SVM) unlike the standard Deep Learning approach. With the proposed technique, they utilised both the precise classification feature of SVM and the automatic feature extraction feature of CNN. The experimental results show that the technique outperforms ensemble-based models with an F1 score of %99.98 and an accuracy of approximately %99.98 [16].

In a study conducted to protect banks from domain name system attacks, a model is proposed in which a system trained with machine learning methods can decide whether to enter the banking network by analysing network packets. Random Forest classifier, Logistic Regression, Support Vector Machine and Gaussian Native Bayes machine learning algorithms were used to train the proposed system. It is stated that the Support Vector Machine method performs better than other methods for detecting domain name server attacks [17].
In another study in the literature, a DNS tunnelling detection method based on cache features was proposed. The proposed method was able to efficiently characterise DNS tunnelling traffic with the given features. In addition, a rule-based filter and a long short-term memory (LSTM) based filter are presented using the proposed features. It is shown that LSTM detects attacks faster, but the rule-based filter detects more DNS tunnelling attacks [18].

DNS tunnelling detection was performed in a study using simple supervised learning schemes applying statistical properties of DNS queries and responses. The emphasis was on detecting small portions of malicious data hidden by DNS communication. Despite the simplicity of the system, it has been shown that good results are achieved by replicating individual detections on successive instances over time and making a global decision through a majority voting scheme
[19].

In a study utilising Software Defined Network (SDN) architecture, the problem of DNS-based data leakage detection and mitigation is addressed. Popular DNS data leakage attacks and existing data leakage detection mechanisms are analysed to create a feature set for DNS data leakage detection. In this study, DNSxD implementation is presented and its performance is evaluated by comparing it with existing intrusion detection mechanisms [20].

This study investigates the effectiveness of machine learning algorithms in detecting DNS tunnelling attacks. Various machine learning algorithms (Random Forest, K-Nearest Neighbour, Decision Trees, Extra Trees) are tested on the CIC-Bell-DNS-EXF-2021 dataset. This study addresses the challenges of detecting DNS tunnelling attacks and proposes an approach that offers higher accuracy and efficiency in this area. Our study aims to contribute to the development of more effective security measures against DNS tunnelling.

2. MATERIAL AND METHOD

2.1. Dataset

The CIC-Bell-DNS-EXF-2021 dataset, available from the UNB website, was used to test the classification algorithms. CIC-Bell-DNS-EXF-2021 is a large dataset created by leaking various file types from small to large. The dataset consists of 323,698 severe attacks, 53,978 mild attacks and 949,711 different benign samples and 30 features extracted from these samples. The dataset was analysed on five consecutive days in two categories, light file attack and heavy file attack, and situational and non-situational categorical features were coded. There are six file types in each category: audio, .exe, compressed, text, image and video. The dataset is considered a valuable resource due to the wide scope of the Domain Name System (DNS), its large volume, controlled environment, diverse range of data, annotated labels, and the fact that the dataset can be used for various purposes such as

training, testing, development, and experimental studies [10]. The statistical distribution of the data in the data set is shown in Figure 1.

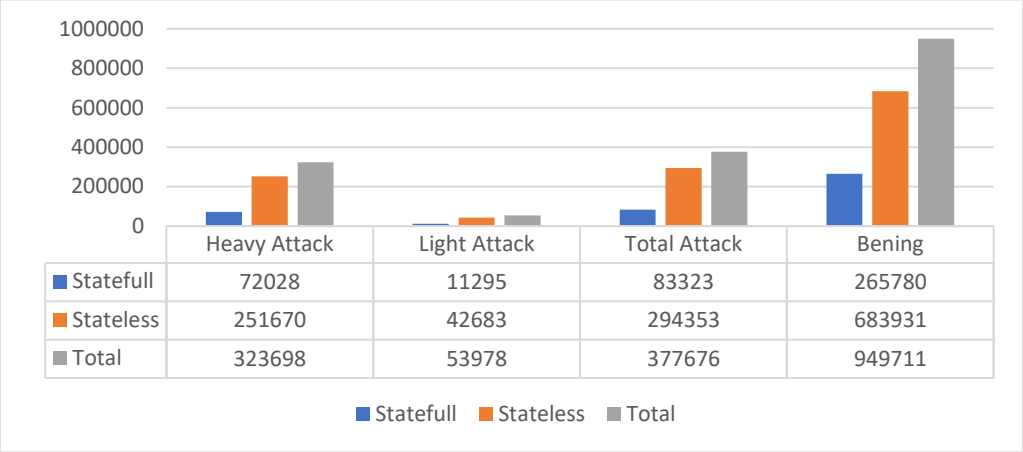

Figure 1. Statistics of the dataset

The 30 attributes in the dataset are given in Table 1 according to their state and non-state characteristics.

Table 1. Data Set Attributes [10]

Feature name	**Description**	**State**
rr_type	The type of resource record, e.g., A, MX, TXT …	stateful
rr_count	The number of entries in each section is as follows: question, answer, authority, and additional.	stateful
rr_name_length	The length of the resource record name.	stateful
rr_name_entropy	The entropy of the resource record name	stateful
rr_type_frequency	The number of packets of a given resource record type for a given domain in relation to the total number of packets for that domain.	stateful
rr	The distribution of A and AAAA resource records, that is to say the rate of A and AAAA records per domain in window τ, is a key metric in this analysis.	stateful
distinct_ns	The number of distinct Name Server (NS) records represents the total number of NSs resolved in the Domain Name System (DNS) Database (DNSDB).	stateful
a_records	The number of distinct A records is defined as the total number of IP addresses resolved in DNSDB.	stateful
unique_country	The domain name of a given country is represented by a distinct country name in the Windows Tau operating system.	stateful
unique_asn	The presence of distinct autonomous system numbers (ASNs) within the specified temporal window (τ) is evident.	stateful
unique_ttl	It is observed that distinct time-to-live (TTL) values are present within the window τ.	stateful
distinct_ip	It is necessary to identify distinct IP values for a given domain at a specific point in time, designated as τ.	stateful
distinct_domains	A domain may be distinguished from other domains that share the same IP address and that resolve to a given domain in window τ.	stateful
reverse_dns	The results of a reverse DNS query for a specific domain at a given point in time are presented in the following section.	stateful
ttl_mean	The mean of the TTL in the window τ.	stateful
ttl_variance	The variability of the time-to-live (TTL) parameter within the specified window.	stateful
FQDN_count	The total number of characters in the Fully Qualified Domain Name (FQDN) is to be determined.	stateless
subdomain_length	The number of characters in the subdomain is to be determined.	stateless
upper	The number of uppercase characters.	stateless
lower	The number of lowercase characters	stateless
numeric	The number of numerical characters	stateless
entropy	Entropy of query name	stateless

special	The number of special characters is defined as the number of characters that do not conform to the standard alphanumeric set. These include the following: Dashes, Underscores, Equals signs, Spaces, Tabs	stateless
labels	The number of labels present in a given query can be determined by examining the query name, which in the case of "www.scholar.google.com" comprises four labels separated by dots	stateless
labels_max	The maximum length of a label	stateless
labels_average	The mean length of a label	stateless
longest_word	The longest meaningful word in a given domain, averaged over the domain length	stateless
sld	The second level domain	stateless
len	The length of the domain and subdomain	stateless
subdomain	It is irrelevant whether the domain in question has a subdomain or not	stateless

The features used in this study were chosen to reflect the specific behaviour of DNS tunnelling attacks. For example, features such as rr_name_entropy and rr_type_frequency were used to understand the distribution and complexity of DNS requests. These features are critical for the detection of anomalies in DNS traffic. Random Forest, K-Nearest Neighbour, K-Nearest Neighbour, Decision Trees and Extra Trees were chosen as classification algorithms because these algorithms are known to give effective results in high dimensional datasets. In particular, Random Forest and Extra Trees are preferred because they provide diversification and stability by using a tree structure in large data sets.

2.2. Machine Learning Techniques and Classification Algorithms

2.2.1. Random Forest

Random forest (RF) classifiers are frequently used in the context of domain name system (DNS) security to identify cyber threats [21]–[23]. These classifiers use data-driven techniques such as information gain and genetic algorithms to analyse DNS queries and identify botnet attacks [21]. When combined with random sampling, RF classifiers can be used to identify malicious actors based on their impact on passive traffic. Significant progress has been made in terms of accurate identification, reducing training time and improving accuracy [22]. Furthermore, RF classifiers have been shown to be effective in distinguishing between legitimate and botnet traffic in distributed denial of service (DDoS) attacks. In terms of its capacity to discriminate on the basis of quality and its own processing time compared to other algorithms, this algorithm has shown superior performance [23]. Overall, RF classifiers play an important role in effectively identifying and mitigating various cyber threats, thereby enhancing DNS security.

2.2.2. K-Nearest Neighbour

K-Nearest Neighbours (KNN) classification is a widely used method in various applications, including intrusion detection systems. The Distance Sum-based K-Nearest Neighbours (DS-kNN) approach, as discussed in [24], improves the original KNN algorithm by computing the distance sum of k-nearest neighbours for improved classification accuracy, detection rate and intrusion classification. Furthermore, a parameter-free KNN classifier proposed in [25] adapts the value of k according to the data distribution, thus eliminating the need for manual parameter tuning. Furthermore, K-nearest neighbour imputation methods such as Fast-K-nearest neighbour imputation (FKNNI) in [26] demonstrate the effectiveness of KNN in rapid disease diagnosis by efficiently handling missing data. Collectively, these studies demonstrate the versatility and effectiveness of KNN techniques in various domains and demonstrate their potential for applications such as DNS tunnel detection.

2.2.3. Decision Trees

Decision tree classifiers have been used in a number of studies to classify DNS tunnels. For example, one study proposed a fusion RF-LR model that combines random forest with logistic regression as the base learner using leaf nodes of decision trees to effectively classify DNS tunnels [13]. Another research project focused on feature reduction using genetic algorithm and decision tree classifier to accurately classify denial of service attacks in the NSL-KDD dataset, thus demonstrating the importance of feature selection in classification tasks [27]. Furthermore, one study highlighted the effectiveness of integrating a decision tree with support vector machines and logistic regression to achieve an impressive %99.96 detection accuracy in identifying DNS tunnels based on key features such as time interval, request packet size, record type and subdomain entropy [28]. These findings collectively demonstrate the important role of decision tree classifiers in effectively detecting and classifying DNS tunnels for cyber security purposes.

2.2.4. Extra Trees

Extra Trees Classification, a variant of Random Forest, is a highly effective technique for detecting DNS tunnels in network traffic. By extracting features from DNS data and training random forest classifiers, the researchers were able to successfully distinguish normal DNS activity from tunnelling activity and detect both trained and untrained tunnels with high accuracy [9]. Another study proposed a fusion RF-LR model combining random forest and logistic regression to classify DNS tunnels based on domain characteristics [13]. This approach showed superior accuracy, recall and stability compared to other algorithms. Furthermore, an effective DNS tunnel detection mechanism was developed using features such as time intervals, packet sizes, record types and subdomain entropy, and an impressive detection accuracy of %99.96 was achieved with Support Vector Machine, Decision Tree and Logistic Regression models [28]. These studies collectively demonstrate the effectiveness of machine learning algorithms, including Extra Tree Classification, in identifying DNS tunnels and improving network security [29].

2.3. Performance Metrics

A confusion matrix is a table used in machine learning to visualise the performance of a classification model. It compares actual and predicted values for different classes and consists of four main components: true positives, true negatives, false positives and false negatives. The matrix is crucial for evaluating the effectiveness of classification models as it provides information about model accuracy, precision, recall and F1 score. To address the complexity of hierarchical classification problems, a hierarchical confusion matrix is introduced. This confusion matrix allows traditional evaluation criteria to be adapted to hierarchical scenarios [30]. Furthermore, alternative approaches such as the Prayatul Matrix have been proposed to compare supervised machine learning models based on individual instances in datasets. The Prayatul Matrix offers a more detailed and insightful analysis compared to traditional confusion matrix-based scores such as accuracy, precision and recall [31].

3. DISCUSSION AND RESULTS

In this study, Random forest, K-nearest neighbour, Decision trees and extra trees classification algorithms were used on the CIC-Bell-DNS-EXF-2021 dataset. GridSearchCV was used to maximise the performance of the classification algorithms. GridSearchCV aims to identify the best combinations through cross-validation over a given grid of hyperparameters. The final hyperparameters were selected by evaluating the model on metrics such as accuracy, F1 score and error rate. For example, for Random Forest, the max_depth parameter was limited (max_depth=5) to prevent the model from overlearning. Other parameters such as min_samples_split and min_samples_leaf were also optimised to increase the accuracy of the model while avoiding overfitting. As a result, these selected parameters are suitable for optimally modelling both stateful and stateless traffic in the dataset. CIC-Bell-DNS-EXF-2021 contains 323,698 heavy attacks, 53,978 light attacks, and 949,711 different benign samples generated by leaking various file types from small to large, and 30 different attributes. The dataset was trained on classification algorithms separately for stateful and stateless traffic. The performance of the algorithms was measured in terms of accuracy.

Pre-processing steps were meticulously performed on the CIC-Bell-DNS-EXF-2021 dataset. Firstly, certain columns in the dataset (e.g. timestamp) were removed to prevent the model from overlearning. These columns may cause the model to overfit to specific instances in the dataset instead of improving its overall performance. Also, NaN values have been replaced with 0, as these cases often imply missing data, and zeroing allows the model to treat these cases in a neutral way. Repetitive rows were also removed from the dataset because such data can be misleading in the learning process of the model. The training and test data were split into 80% training and 20% testing to maximise the diversity and generalisability of the dataset. Table 2 shows the numerical attributes of the traffic with and without state before training.

Table 2. Attributes Used

Stateful Attributes	rr, A_frequency, NS_frequency, CNAME_frequency, SOA_frequency, NULL_frequency, PTR_frequency, HINFO_frequency, MX_frequency, TXT_frequency, AAAA_frequency, SRV_frequency, OPT_frequency, rr_count, rr_name_entropy, rr_name_length, distinct_ns, a_records, ttl_mean, ttl_variance, label
Stateless Attributes	FQDN_count, subdomain_length, upper, lower, numeric, entropy, special, labels, labels_max, labels_average, len, subdomain, label

As a result of hyperparameter optimisation with GridSearchCV for stateful and stateless DNS traffic, the parameters in Table 3 were found.

Table 3. Hyperparameters for Stateful and Stateless DNS Traffic

Machine Learning Algorithm	**Parameters (Stateful)**
Random Forest	{'max_depth':5, 'max_features':'log2', 'min_samples_leaf':4, 'min_samples_split': 10, 'n_estimators': 50}
K-Nearest Neighbour	{'metric': 'manhattan', 'n_neighbors': 4, 'weights': 'uniform'}
Decision Trees	{'criterion':'gini', 'max_depth':2, 'min_samples_leaf':1, 'min_samples_split':2}
Extra Trees	{'max_depth' : 'None', 'min_samples_leaf' : 4, 'min_samples_split' : 10, 'n_estimators' : 50}
Machine Learning Algorithm	**Parameters (Stateless)**
Random Forest	{'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 50}
K-Nearest Neighbour	{'metric': 'euclidean', 'n_neighbors': 10, 'weights': 'distance'}
Decision Trees	{'criterion':'gini', 'max_depth':2, 'min_samples_leaf':1, 'min_samples_split':2}
Extra Trees	{'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 50}

As a result of the classification with the hyperparameters found for the algorithms, K-nearest neighbour, Decision trees, Random forest and Extra trees achieved %99.68, %99.74, %99.71 and %99.74 accuracy rates for DNS traffic with state. For stateless DNS traffic, K-nearest neighbour %93.94, Decision trees %93.95, Random forest %93.95 and Extra trees %93.95. The complexity matrices of the classification algorithms are given in Figure 2 and Figure 3.

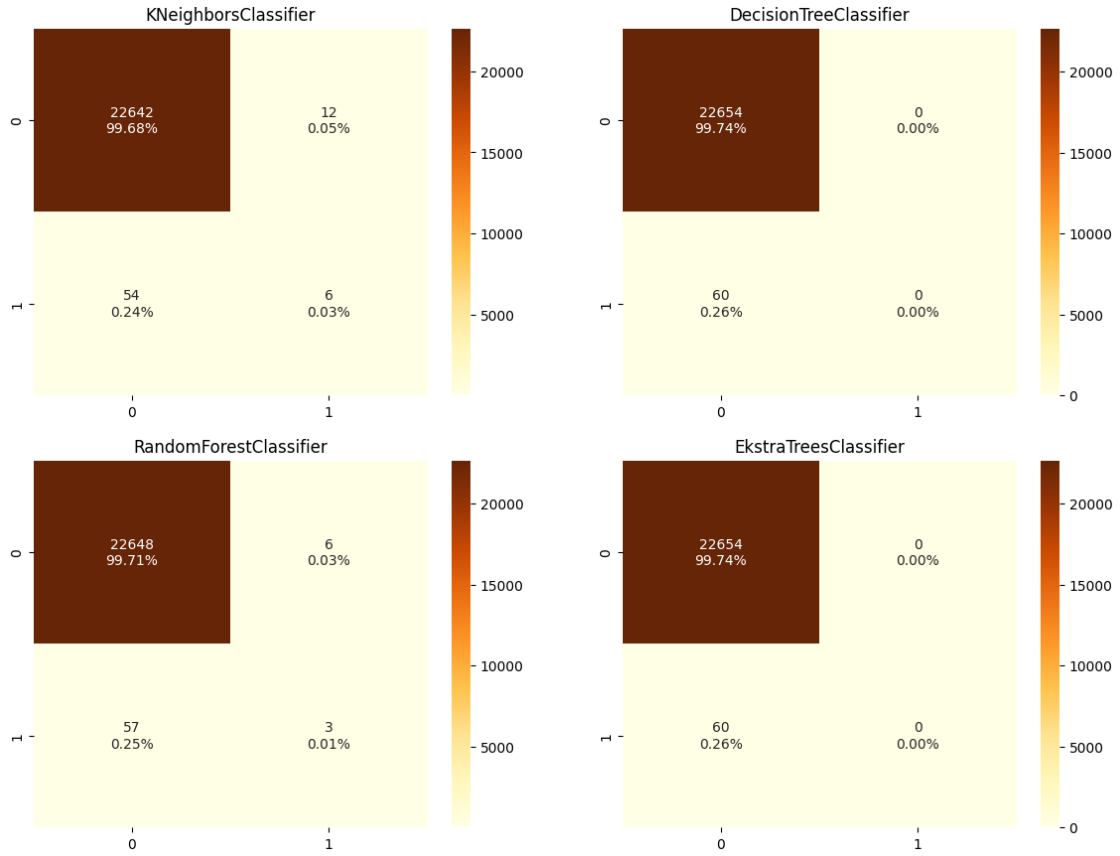

Figure 2. Confusion matrices of classification algorithms with stateful traffic

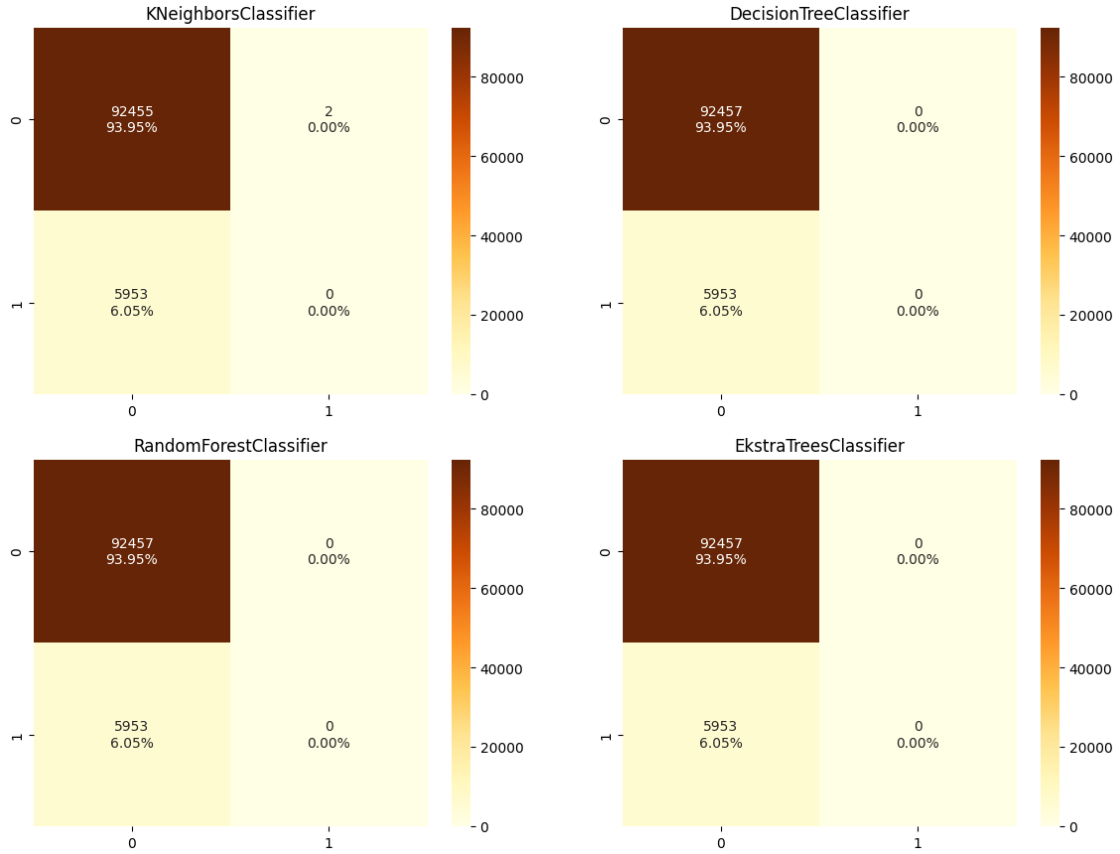

Figure 3. Confusion matrices of classification algorithms with stateless traffic

Table 4 shows the accuracy values obtained for stateful and stateless traffic.

Table 4. Accuracy Rates of Algorithms

Algoritma	**Statefull**	**Stateless**
Random Forest	%99.71	%93.95
K-Nearest Neighbour	%99.68	%93.95
Decision Trees	%99.74	%93.95
Extra Trees	%99.74	%93.95

4. CONCLUSION

The research presented in this paper presents an evaluation of machine learning algorithms for the detection of DNS tunnelling attacks using the CIC-Bell-DNS-EXF-2021 dataset. Experimental results show that machine learning algorithms, in particular Random Forest, K-Nearest Neighbour, Decision Trees and Extra Tree classifiers, exhibit high accuracy in identifying both stateful and stateless DNS traffic anomalies. Hyperparameter optimisation via GridSearchCV further improves the performance of these models.

For stateful traffic, the accuracy of the classifiers is very high, with Decision Trees and Extra Trees achieving the best results with 99.74%. Stationary traffic also showed respectable, albeit slightly lower, accuracy rates, with all tested algorithms converging at around 93.95%. This difference highlights the inherent complexity of detecting DNS tunnelling in stateless traffic compared to stateful traffic.

The findings underline the critical role of feature selection and hyperparameter tuning in optimising machine learning models for cyber security applications. The use of comprehensive datasets such as CIC-Bell-DNS-EXF-2021 allows for robust evaluation of detection techniques, ensuring that models are both effective and reliable.

In conclusion, this study highlights the potential of machine learning algorithms to improve DNS security by effectively detecting DNS tunnelling attacks. The high accuracy rates obtained demonstrate the applicability of these models in real-world scenarios, thus contributing to stronger cybersecurity defences against sophisticated

data exfiltration methods. Future research could focus on integrating these models into broader security frameworks and exploring their performance in different network environments.

REFERENCES

[1] O. Abualghanam, H. Alazzam, B. Elshqeirat, M. Qatawneh, ve M. A. Almaiah, "Real-Time Detection System for Data Exfiltration over DNS Tunneling Using Machine Learning", *Electron. 2023, Vol. 12, Page 1467*, c. 12, sayı 6, s. 1467, Mar. 2023, doi: 10.3390/ELECTRONICS12061467.

[2] Y. Ozery, A. Nadler, ve A. Shabtai, "Information-Based Heavy Hitters for Real-Time DNS Data Exfiltration Detection and Prevention", Tem. 2023, Erişim: 14 Haziran 2024. [Çevrimiçi]. Available at: https://arxiv.org/abs/2307.02614v1

[3] S. Sugasawa, Y. Shibahashi, H. Kunimune, H. Goromaru, ve S. Tanimoto, "DNS-tunneling-detection Method by Monitoring DNS Subdomain Length for General Usage", ss. 121–122, Oca. 2023, doi: 10.1109/GCCE56475.2022.10014255.

[4] L. Salat, M. Davis, ve N. Khan, "DNS Tunnelling, Exfiltration and Detection over Cloud Environments", *Sensors 2023, Vol. 23, Page 2760*, c. 23, sayı 5, s. 2760, Mar. 2023, doi: 10.3390/S23052760.

[5] L. De Souza Bezerra Borges, R. De Oliveira Albuquerque, ve R. T. De Sousa Junior, "A security model for DNS tunnel detection on cloud platform", *2022 Work. Commun. Networks Power Syst. WCNPS 2022*, 2022, doi: 10.1109/WCNPS56355.2022.9969715.

[6] W. Ellens, P. Zuraniewski, A. Sperotto, H. Schotanus, M. Mandjes, ve E. Meeuwissen, "Flow-Based Detection of DNS Tunnels", *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, c. 7943 LNCS, ss. 124–135, 2013, doi: 10.1007/978-3-642-38998-6_16.

[7] C. M. Lai, B. C. Huang, S. Y. Huang, C. H. Mao, ve H. M. Lee, "Detection of DNS Tunneling by Feature-Free Mechanism", *DSC 2018 - 2018 IEEE Conf. Dependable Secur. Comput.*, Oca. 2019, doi: 10.1109/DESEC.2018.8625166.

[8] Y. Shao, X.-D. Li, A. F. Sani, ve M. A. Setiawan, "DNS tunneling Detection Using Elasticsearch", *IOP Conf. Ser. Mater. Sci. Eng.*, c. 722, sayı 1, s. 012064, Oca. 2020, doi: 10.1088/1757-899X/722/1/012064.

[9] A. L. Buczak, P. A. Hanke, G. J. Cancro, M. K. Toma, L. A. Watkins, ve J. S. Chavis, "Detection of tunnels in PCAP data by random forests", *Proc. 11th Annu. Cyber Inf. Secur. Res. Conf. CISRC 2016*, Nis. 2016, doi: 10.1145/2897795.2897804.

[10] S. Mahdavifar *vd.*, "Lightweight Hybrid Detection of Data Exfiltration using DNS based on Machine Learning", *ACM Int. Conf. Proceeding Ser.*, ss. 80–86, Ara. 2021, doi: 10.1145/3507509.3507520/SUPPL_FILE/P80-MAHDAVIFAR-SUPPLEMENT.PPTX.

[11] S. Wang, L. Sun, S. Qin, W. M. Li, ve W. Liu, "KRTunnel: DNS channel detector for mobile devices", *Comput. Secur.*, c. 120, s. 102818, Eyl. 2022, doi: 10.1016/J.COSE.2022.102818.

[12] R. Mitsuhashi, Y. Jin, K. Iida, T. Shinagawa, ve Y. Takai, "Malicious DNS Tunnel Tool Recognition Using Persistent DoH Traffic Analysis", *IEEE Trans. Netw. Serv. Manag.*, c. 20, sayı 2, ss. 2086–2095, Haz. 2023, doi: 10.1109/TNSM.2022.3215681.

[13] X. D. Li, Y. F. Song, ve Y. Q. Li, "DNS Tunnel Detection Scheme Based on Machine Learning in Campus Network", *Proc. - 2022 4th Int. Conf. Mach. Learn. Big Data Bus. Intell. MLBDBI 2022*, ss. 253–257, 2022, doi: 10.1109/MLBDBI58171.2022.00056.

[14] O. Abualghanam, H. Alazzam, B. Elshqeirat, M. Qatawneh, ve M. A. Almaiah, "Real-Time Detection System for Data Exfiltration over DNS Tunneling Using Machine Learning", *Electron. 2023, Vol. 12, Page 1467*, c. 12, sayı 6, s. 1467, Mar. 2023, doi: 10.3390/ELECTRONICS12061467.

[15] B. Wang, G. Xiong, G. Gou, J. Song, Z. Li, ve Q. Yang, "Identifying DoH Tunnel Traffic Using Core Feathers and Machine Learning Method", *Proc. 2023 26th Int. Conf. Comput. Support. Coop. Work Des. CSCWD 2023*, ss. 814–819, 2023, doi: 10.1109/CSCWD57460.2023.10152678.

[16] A. Lal, A. Prasad, A. Kumar, ve S. Kumar, "DNS-Tunnet: A Hybrid Approach for DNS Tunneling Detection", *CTISC 2022 - 2022 4th Int. Conf. Adv. Comput. Technol. Inf. Sci. Commun.*, 2022, doi: 10.1109/CTISC54888.2022.9849774.

[17] A. Khan ve I. Sharma, "AI-Enabled Approach for Preventing DNS Attacks on Banking Institutions", *2023 IEEE Int. Conf. Res. Methodol. Knowl. Manag. Artif. Intell. Telecommun. Eng. RMKMATE 2023*, 2023, doi: 10.1109/RMKMATE59243.2023.10369196.

[18] N. Ishikura, D. Kondo, V. Vassiliades, I. Iordanov, ve H. Tode, "DNS Tunneling Detection by Cache-Property-Aware Features", *IEEE Trans. Netw. Serv. Manag.*, c. 18, sayı 2, ss. 1203–1217, Haz. 2021, doi: 10.1109/TNSM.2021.3078428.

[19] M. Aiello, M. Mongelli, ve G. Papaleo, "Basic classifiers for DNS tunneling detection", *Proc. - IEEE Symp. Comput. Commun.*, ss. 880–885, 2013, doi: 10.1109/ISCC.2013.6755060.

[20] J. Steadman ve S. Scott-Hayward, "DNSxD: Detecting Data Exfiltration over DNS", *2018 IEEE Conf. Netw. Funct. Virtualization Softw. Defin. Networks, NFV-SDN 2018*, Kas. 2018, doi: 10.1109/NFV-

SDN.2018.8725640.

[21] A. Moubayed, M. N. Injadat, ve A. Shami, "Optimized Random Forest Model for Botnet Detection Based on DNS Queries", *Proc. Int. Conf. Microelectron. ICM*, c. 2020-December, Ara. 2020, doi: 10.1109/ICM50269.2020.9331819.

[22] A. Dickson ve C. Thomas, "ATTACK DETECTION AVAILING FEATURE DISCRETION USING RANDOM FOREST CLASSIFIER", *Comput. Sci. Eng. An Int. J.*, c. 12, sayı 6, 2022, doi: 10.5121/cseij.2022.12611.

[23] Z. F. Faruq, T. Mantoro, M. A. Catur Bhakti, ve Wandy, "Random Forest Classifier Evaluation in DDoS Detection System for Cyber Defence Preparation", *2022 IEEE 8th Int. Conf. Comput. Eng. Des. ICCED 2022*, 2022, doi: 10.1109/ICCED56140.2022.10010341.

[24] R. Taguelmimt ve R. Beghdad, "DS-kNN: An Intrusion Detection System Based on a Distance Sum-Based K-Nearest Neighbors", *https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/IJISP.2021040107*, c. 15, sayı 2, ss. 131–144, Oca. 1M.S., doi: 10.4018/IJISP.2021040107.

[25] D. S. Jodas, L. A. Passos, A. Adeel, ve J. P. Papa, "PL-k NN: A Parameterless Nearest Neighbors Classifier", *Int. Conf. Syst. Signals, Image Process.*, c. 2022-June, 2022, doi: 10.1109/IWSSIP55020.2022.9854445.

[26] D. Chen, R. Ma, ve H. Du, "A fast incomplete data classification method based on representative points and K-nearest neighbors", *2022 IEEE Conf. Telecommun. Opt. Comput. Sci. TOCS 2022*, ss. 423–428, 2022, doi: 10.1109/TOCS56154.2022.10016185.

[27] D. Wilborne, "Application of Decision Tree Classifier in Detection of Specific Denial of Service Attacks with Genetic Algorithm Based Feature Selection on NSL-KDD", Eki. 2022, Erişim: 27 Haziran 2024. [Çevrimiçi]. Available at: https://arxiv.org/abs/2210.10232v1

[28] J. Liu, S. Li, Y. Zhang, J. Xiao, P. Chang, ve C. Peng, "Detecting DNS Tunnel through Binary-Classification Based on Behavior Features", içinde *2017 IEEE Trustcom/BigDataSE/ICESS*, 2017, ss. 339–346. doi: 10.1109/Trustcom/BigDataSE/ICESS.2017.256.

[29] S.-Y. Zhang, F.-T. Zou, L.-H. Wang, ve M. Chen, "Detecting DNS-based covert channel on live traffic", *J. China Inst. Commun.*, c. 34, sayı 5, ss. 143–151, 2013.

[30] K. Riehl, M. Neunteufel, ve M. Hemberg, "Hierarchical confusion matrix for classification performance evaluation", *J. R. Stat. Soc. Ser. C Appl. Stat.*, c. 72, sayı 5, ss. 1394–1412, Ara. 2023, doi: 10.1093/JRSSSC/QLAD057.

[31] A. Biswas, "Prayatul Matrix: A Direct Comparison Approach to Evaluate Performance of Supervised Machine Learning Models", Eyl. 2022, Erişim: 27 Haziran 2024. [Çevrimiçi]. Available at: https://arxiv.org/abs/2209.12728v1