Models for Evaluation Intrusion Detection Systems in IoT

Subjects: Computer Science, Cybernetics | Computer Science, Artificial Intelligence

Contributor: RUBAYYI Alghamdi , Martine Bellaiche

Using the Internet of Things (IoT) for various applications, such as home and wearables devices, network applications, and even self-driven vehicles, detecting abnormal traffic is one of the problematic areas for researchers to protect network infrastructure from adversary activities. Several network systems suffer from drawbacks that allow intruders to use malicious traffic to obtain unauthorized access. Attacks such as Distributed Denial of Service attacks (DDoS), Denial of Service attacks (DoS), and Service Scans demand a unique automatic system capable of identifying traffic abnormality at the earliest stage to avoid system damage. Numerous automatic approaches can detect abnormal traffic. However, accuracy is not only the issue with current Intrusion Detection Systems (IDS), but the efficiency, flexibility, and scalability need to be enhanced to detect attack traffic from various IoT networks.

IoT network cyberattack machine learning IDS

1. Introduction

Internet of Things (IoT) is a network that uses unique identifiers (UIDs) to connect things, including but not limited to computing devices, automatic and digital tools, objects, creatures, and people with the capacity to share data without needing human-to-human or human-to-computer interaction. IoT connects the physical and virtual worlds in simple terms. The fundamental idea behind IoT is to establish a safe, self-governing connection that allows data to be exchanged between real-world physical objects and apps [1]. The IoT Analytics report indicates that more than 11 billion IoT devices are connected and used in current lives. Moreover, it is shown that there is an increase of more than 10% of IoT devices. The expectation is that in 2025, there will be around 21 billion IoT devices connected around the world [1]. As a result, the IoT has experienced rapid expansion in recent years due to widespread adoption in various sectors and industries, including smart cities and smart homes, agriculture, transportation, logistics, and healthcare [2]. The integration of IoT with growing and rapidly rising software such as Artificial Intelligence (AI), big data, and 5G is also receiving much attention. However, as the world becomes more interconnected through the Internet and the IoT ecosystem grows in size, application, connectivity, and complexity, the need to adequately secure IoT components at all levels, from physical to user, becomes more vital than ever. As a consequence, network security becomes essential. Advanced hacking techniques can compromise the security of IoT communications and computational components. There are numerous attacks in IoT networks, including but not limited to Denial of Service (DoS) and Distributed DoS (DDoS). This type of attacks can make the device unreachable to handle actual demands by flooding and over-burdening the IoT device with pointless and superfluous messages. The other type of attack is the Replay attack. This attack can be executed by capturing the traffic to distinguish the unencrypted information, which works with hackers to accomplish Man-in-the-Middle (MITM). Additionally, a spoofing attack where attackers can take on the appearance of legitimate clients and forward fake Global Positioning System (GPS) information to the nodes. Therefore, IDS is crucial to protect IoT from these attacks and secure the network by detecting abnormal behaviours using artificial intelligence [3]. IDS is usually divided into two types: anomaly and misuse methods that depend on being signature-based. Various types of machine and deep learning models for IDS have been published and trained to detect essential and specific threats for conventional networks and partly in the Internet of Things networks [4].

2. Problem Definition

The IoT network is heterogeneous because of the various applications and devices involved. According to the literature, detecting attacks is a classification issue since the purpose is to determine if the packet is legitimate or malicious, which solves 50% of the problem. However, as an IoT network is complicated, knowing the type of attack on the network is essential; otherwise, scans would be time-consuming, giving attackers more accessible access to obtain or modify information and even destroying the system. Moreover, there are no standardized validation efforts for intrusion detection in IoT. The validation strategy is critical to evaluate the model under different factors such as different network parameters and configurations, different datasets and different scenarios. The dataset is crucial to learn about the correctness of a model. However, traditional IDS in several works has been validated based on data from the experiments performed, which is an outdated dataset that does not have the latest attacks in IoT. There are also some challenges and problems in successfully deploying an anomaly system in the real-world environment. The challenges include high False Positive Rates (FPR), where the system labelled the normal packets as abnormal packets because of the lack of labelled data and low Detection Rates (DR). The high velocity, complexity, and variability in IoT networks produce a high traffic volume from several networks that need to be extracted, aggregated, and processed.

The objective is to develop ensemble detection model to achieve a reliable, scalable, adaptable, flexible, and speedy IDS for heterogeneous IoT networks that predicts if an attack happens in real-time and, if so, determine its type in order to enable administrators to speed up the recovery of their systems since the architecture will help to reduce scans time to know the type and source of the attack. Moreover, producing heterogeneous networks makes learning several traffic patterns challenging for an individual model. Therefore, to develop classification accuracy, increase decision-making, and decrease selection of a weaker classifier, an aggregate model that consolidates several models is the solution.

3. Single IDS

3.1. UNSW-NB15 Dataset

Aleesa et al. [5] proposed an IDS using deep learning methods. The researchers utilized a popular dataset called UNSW-NB15 to evaluate their approach performance for binary and multi-class classification. The researchers consider accuracy as the primary performance metric. The result shows that the accuracy obtained is 99.26% and 97.89% for binary and multi-class classification. However, They compared the performance of the proposed models with other published works and claimed that these results were outstanding to current ones.

Zhou et al. [6] proposed a feature embeddings model called DFEL. The researchers trained a neural network model to reduce the features matrix and construct embeddings. Then several classifier models, such as Support Vector Machine (SVM), Decision Trees (DT), K-Nearest Neighbor (KNN), and GaussianNB used to classify traffic. To evaluate the performance of the model, theresearchersused UNSW-NB15. However, because the used dataset is more sophisticated and reflects real-world traffic, the high-performance model reached 91.22% accuracy.

3.2. Aposemat IoT-23 Dataset

PuTianet et al. [7] present DC-Adam technique. The idea is based on a federated learning-based detection method. Asynchronous update technique enhances the stability of the model by decreasing the impacts from the stragglers, which leads to improved model accuracy. The researchers evaluate the effectiveness of the suggested strategy corresponding to existing state-of-the-art strategies using the publicly available dataset IoT-23. The result of the proposed method is 89.50%.

Nukavarapu et al. [8] uses Generative adversarial networks (GANs) to construct a multistage classifier model. The proposed model automatically extracts features from labelled data. The researchers used IoT23 labelled data during the training to evaluate their model. The result received is 92% of accuracy.

3.3. Ensemble IDS

3.3.1. UNSW-NB15 Dataset

Ahmad et al. [9] introduced a technique to detect and remove suspicious packets from the Internet of Things. Random Forest (RF), Support Vector Machine (SVM), and Artificial Neural Networks (ANN) were used by the researchers as supervised machine learning techniques. The researchers utilize the full features related to Transmission Control Protocol (TCP) and Message Queuing Telemetry Transfer(MQTT) in the UNSW-NB15 dataset. The proposed model reached 98.67% and 97.37% accuracy in binary and multiclass classification, accordingly.

Rashid et al. [10] present a stacking ensemble approach. In the first phase, the researchers reduce the dimensionality of the features by applying the best feature selection method. In the second step, an ensemble approach employs for the detection process. Their model utilizes three different models called Decision Trees (DT), Random XGBoost, and Random Forest (RF). The results showed that the proposed model reached 94% accuracy on the UNSW-NB15 dataset.

Moustafa et al. [11] proposed an ensemble method composed of Artificial neural network (ANN), Naive Bayes (NB), and Decision Tree (DT). The model goal is to detect botnet attacks and suspicious activity in IoT. The results showed that their ensemble technique reached an accuracy of 98.97% using statistical flow features of Hypertext Transfer Protocol (HTTP) and Message Queuing Telemetry Transfer(MQTT) generated from UNSW-NB15.

Smitha et al. [12] proposed a stacking ensemble-based IDS. The researchers built base classifiers utilizing Linear Regression (LR), K-Nearest Neighbour (KNN), and Random Forest (RF). For the meta-classifier, they used a Support Vector Machine (SVM). Experimentations have been performed on UNSW-NB15 and UGR'16 heterogeneous datasets. The outcomes indicated that the proposed model reached 94% of accuracy on the UNSW-NB15 dataset.

3.3.2. Aposemat IoT-23 Dataset

Dutta et al. [13] developed an ensemble model including deep neural network (DNN) and Long Short Term Memory (LSTM) for detecting and classifying anomalies. Researchers conducted individual testing on three heterogeneous datasets; and determined that the proposed technique outperformed individual and meta-classifiers such as Support Vector Machine (SVM) and Random Forest (RF). The dataset used is IoT-23. 98% of F1 scores were achieved, with an accuracy of 99.7%.

Amiya et al. [14] proposed a security architecture and attack detection method based on Long Short Term Memory (LSTM) and Convolutional Neural Network (CNN). According to the model's results, 96% of attacks are detected correctly.

3.3.3. Ton_IoT Dataset

Booij et al. [15] presented a smart IDS that is suitable for the industrial field. Several classifiers were used in the experiments, including the Random Forest (RF), Gradient Boosting Machine (GBM), and multilayer perceptron (MLP). A 98.07% performance was obtained when the proposed approach was applied to the ToN_IoT dataset.

Alsaedi et al. [16] introduced the ToN_IoT dataset, a current dataset for developing and testing IDSs. Many machine learning approaches were applied such as Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), Support Vector Machine (SVM), Naive Bayes (NB), and K-Nearest Neighbor (KNN). CART model achieved 87% of accuracy on the weather dataset.

4. Big Data

4.1. Big Data Processing Systems

Spark is considered one type of big data processing tool. This tool is one of the core components of every big data architecture. Spark analyzes stored data in repository systems or data consumed systems. Apache Spark was founded in 2009 by AMPLab at UC Berkeley. It is fitting for a wide scope of use cases. Generally, four libraries built on top of the Spark processing engine are Spark SQL, Spark Streaming, MLlib for machine learning algorithms, and GraphX for graph computations. The programming languages supported by Spark are Java, Python, Scala, and R. It is possible to run Spark using its cluster on Hadoop YARN, Mesos or Kubernetes. Spark supports both batch and stream processing [17][18].

4.2. Big Data Storage Systems

Analyzing massive amounts of heterogeneous data is unattainable with conventional relational databases such as SQL. In terms of storing big data, scalable, fault-tolerant, and reliable systems such as NoSQL databases or distributed databases are suitable. For that reason Hadoop Distributed File System (HDFS) is chosen. HDFS is a big data technology designed to run on low-cost hardware with fault-tolerance. HDFS consists of Name Nodes and Data Nodes. Name Nodes are responsible for how data are parsed and distributed among servers, while Data Nodes are responsible for storing blocks of data. Name Nodes controlled Data Nodes that exist on multiple servers [17][19]. HDFS provides great scalability, fault-tolerance, and reliability.

5. Deep Learning

5.1. Long Short Term Memory (LSTM)

LSTM is a deep learning model that extends Recurrent Neural Networks (RNNs) by enclosing the idea of gates into its units. An essential issue with (RNNs) is their inability to maintain context information when the gradient disappears over a lengthy period [20]. The LSTM approach, on the other hand, eliminates the gradient disappearance issue and allows for longer retention of context information. LSTM is suitable for sequential data, such as in the case of DoS and DDoS. A standard neural network unit only consists of the input activation and the output activation which are related by an activation function. The LSTM models applies the input activation and multiplies the result by some factors. Then the inner activation value of the previous time step, multiplied by the some quantity is added due to the recurrent self-connection. The result is then fed to the activation function after scaling.

5.2. Artificial Neural Network (ANN)

ANNs have recently received significant attention in many areas of study such as engineering, medical science, and economics for a wide diversity of applications. They have been used in classification problems, such as recognizing speech and predicting heart problems in patients. An ANN is a set of single computational nodes that are highly interconnected. The structure of ANN consists of input layers where the number of neurons of the input layer equals the number of features in the dataset, the hidden layers (one or more hidden layers) to compute the weights and then pass them to the output layer for either binary or multi-classification. The over-fitting problem could accrue when there are too many hidden layers. On the other hand, if there are few hidden layers, the cost of training time will increase [21].

5.3. Convolutional Neural Network (CNN)

CNN is a well-known, widely-used structure in deep learning proposed by LeCun et al. [22]. CNN has gained capability in applications and is now capable of facial recognition, text recognition, medical and image classification [23]. It generally consists of three parts: an input layer, a hidden layer, and a fully connected output layer. CNN can be the right solution for creating a high-quality classifier by extraction of the relationship between different events. Moreover, CNN requires fewer parameters with the same depth of network compared with other deep networks, which reduces complexity and speeds the learning process [24].

6. Machine Learning Models

6.1. Random Forest (RF)

It is a supervised learning algorithm. In a random forest, a decision tree is built from a sample of data taken at random, predictions are obtained from each tree, and a vote chooses the best answer. In addition, it serves as a reliable indicator of the feature's relevance. Random forest refers to the collection of these decision trees, called a Forest. An attribute selection indicator such as information gain, gain ratio, and Gini index for each attribute creates the decision trees. To classify the trees, a random sample is used for each tree, and then the most popular class is selected based on the results of the individual tree votes [25].

6.2. AdaBoost

A decision tree is enhanced with the aid of weak learners, or stumps, which each have one node and two leaves [26]. Classification is made possible by AdaBoost's use of several stumps. As the preceding stump makes errors, so does the next one, and so on. Weights are allocated to each sample in the database. After generating the first stump, these weights will vary to guide the creation of the second stump.

6.3. Decision Tree (DT)

A decision tree is a straightforward model for classifying data. It is supervised machine learning where the data are continuously split according to a specific parameter. The components of a decision tree are:

- Nodes: Check the value of a particular feature.

- Edges/Branch: Connect to the next node or leaf based on the results of a test.

- Leaf nodes/Terminal nodes: The nodes produce the final predict result.

The decision variable is categorical/discrete for classification tasks. A procedure called binary recursive partitioning is used to create such a tree. This method involves separating the data into divisions and then further splitting it up on each branch. The data are divided using the decision method based on the characteristic that yields the most information gain (IG) (reduction in uncertainty towards the final decision). This splitting operation is performed at each child node iteratively until the leaves are pure [27].

6.4. K-Nearest Neighbor (KNN)

Classification issues are the most common application for the k-Nearest Neighbor method, which comes under the category of supervised learning. KNN predicts a class or continuous value for a new data point based on the K nearest neighbors (data points). Nearest neighbors are those data points closest in feature space to latest data point, defined by K nearest neighbors. Each unique data point looks for the k closest neighbors and then gives a class label to all of those K Nearest Neighbors, then used as a predicted class for the newly data point [28].

6.5. Naive Bayes

In binary and multi-class classification issues, Naive Bayes is a typical technique. The fundamental concept is that each feature contributes equally and independently to the outcome. It is assumed that each result has a conditional probability. When dealing with categorical input variables, it does better than when dealing with numerical input variables. Classifying the test data is a rapid and straightforward process. The Naive Bayes classifier performs better and requires smaller training examples than other models such as logistic regression [29].

6.6. Light Gradient Boosting Machine (LightGBM)

LGBM and XGBoost have pretty different approaches to tree growth. LGBM’s goal is to make training faster and use less memory while maintaining high accuracy. LGBM Leaf nodes are separated using a histogram-based technique, which significantly impacts memory and efficiency. LGBM may lead to more significant gains in accuracy with each iteration since leaf-wise tree development increases model complexity. However, it also implies high risks of over-fitting. Two unique approaches, Gradient-Based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), are used in the LightGBM algorithm to make it a fast, efficient, and reliable model [30].

6.7. CatBoost

It’s a Gradient Boosting-based algorithm. It has few features optimization parameters and enables faster training and testing. It employs the Ordered Boosting technique, which increases the model’s generalization. Boost improves random forest and other gradient boosting algorithms. It works well with categorical data structures, obviating the requirement for data transformation in the middle [31].

7. Attack Classes

7.1. Backdoor Attacks

These are the type of attacks where hackers try gaining and retaining access to a computer system by bypassing security and authentication by hiding deceitfully [32]. A lot of research has been carried out to detect these attacks. Moreover, special signatures do exist for these attacks that help in detecting and analyzing their behaviour preferably using deep learning techniques.

7.2. Denial of Service Attacks (DoS)

In this attack a single attacker bombards the target with a large amount of data packets (for any protocol e.g., HTTP etc.) to consume its resources with this illegitimate traffic making it unavailable for legitimate ones. A lot of research has been carried out to detect such attacks using simple machine learning and deep learning approaches [33].

7.3. Distributed Denial of Service Attacks (DDoS)

These attacks are way more complex as compared to simpel DoS as in this case a large number of attackers send large amount of packets to a single target making it very hard to detect and stop. Lately, deep learning and Artificial Intelligence based approaches are proven out to detect these attacks with high accuracy [34].

7.4. Injection Attacks

These attacks encompass a wide variety of sub-categories where an attacker supplies a target with untrusted data or commands to subvert the built-in security and achieve desired malicious results. Injection attacks are more common in web applications where attacker supply unwarranted SQL statements which get processed by the target website resulting in exploitation. Machine learning and deep learning based solutions have proven efficient in combating these attacks as rule based Web Application Firewalls (WAFs) show less accuracy [35].

7.5. Man in the Middle Attacks (MiTM)

In these attacks attacker sits between two communicating parties unlawfully and relays the communication of messages between them without their knowledge and consent [36].

7.6. Password Attacks

These attacks aim at subverting password based authentication using any means ranging from cryptanalysis, hash matching or using social engineering techniques. There are many software solutions available which aid the process of password cracking and hash matching. These attacks can be categorized through special signatures and the use of such cracking solutions [37].

7.7. Ransomware Attacks

These attacks make the user data unavailable by encrypting it only to be rehabilitated once a hefty ransom is paid. These attacks have grown rapidly in the recent past and many machine and deep learning based techniques have been employed for their early detection and remediation [38].

7.8. Scanning and Enumeration Attacks

These are basically the preattack steps attackers take in order to gain information about open ports, services running, vulnerable applications and enumerating users and groups etc. in the target system or network. This information later helps in maintaining and retaining access to the victim system. Attackers generally use tools such as NMAP and HPING for scanning and enumeration but the smart attackers can come up with custom scripts. Detection o these scanning payloads is important as they curb the attack before actually having it materialized [39].

7.9. Cross Site Scripting Attacks (XSS)

These attacks are specifically for web applications where the attacker uploads a malicious JavaScript to a remote website so that anyone visiting the website gets infected with it when his client (broswer) executes that JavaScript. Machine and Deep learning based solutions have been used to detect such attacks with high accuracy and precision [40].

References

1. Chopra, K.; Gupta, K.; Lambora, A. Future internet: The internet of things-a literature review. In Proceedings of the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 14–16 February 2019; pp. 135–139.
2. Apostol, I.; Preda, M.; Nila, C.; Bica, I. IoT Botnet Anomaly Detection Using Unsupervised Deep Learning. Electronics 2021, 10, 1876.
3. Levina, A.I.; Dubgorn, A.S.; Iliashenko, O.Y. Internet of things within the service architecture of intelligent transport systems. In Proceedings of the 2017 European Conference on Electrical Engineering and Computer Science (EECS), Bern, Switzerland, 17–19 November 2017; pp. 351–355.
4. Ashraf, J.; Bakhshi, A.D.; Moustafa, N.; Khurshid, H.; Javed, A.; Beheshti, A. Novel Deep Learning-Enabled LSTM Autoencoder Architecture for Discovering Anomalous Events From Intelligent Transportation Systems. IEEE Trans. Intell. Transp. Syst. 2021, 22, 4507–4518.
5. Aleesa, A.; Younis, M.; Mohammed, A.A.; Sahar, N. Deep-intrusion detection system with enhanced UNSW-NB15 dataset based on deep learning techniques. J. Eng. Sci. Technol. 2021, 16, 711–727.
6. Zhou, Y.; Han, M.; Liu, L.; He, J.S.; Wang, Y. Deep learning approach for cyberattack detection. In Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Honolulu, HI, USA, 15–19 April 2018; pp. 262–267.
7. Tian, P.; Chen, Z.; Yu, W.; Liao, W. Towards Asynchronous Federated Learning Based Threat Detection: A DC-Adam Approach. Comput. Secur. 2021, 108, 102344.

8. Nukavarapu, S.K.; Nadeem, T. Securing Edge-based IoT Networks with Semi-Supervised GANs. In Proceedings of the 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), Kassel, Germany, 22–26 March 2021; pp. 579–584.
9. Ahmad, M.; Riaz, Q.; Zeeshan, M.; Tahir, H.; Haider, S.A.; Khan, M.S. Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set. EURASIP J. Wirel. Commun. Netw. 2021, 2021, 10.
10. Rashid, M.; Kamruzzaman, J.; Imam, T.; Wibowo, S.; Gordon, S. A tree-based stacking ensemble technique with feature selection for network intrusion detection. Appl. Intell. 2022.
11. Moustafa, N.; Turnbull, B.; Choo, K.K.R. An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. IEEE Internet Things J. 2018, 6, 4815–4830.
12. Rajagopal, S.; Kundapur, P.P.; Hareesha, K.S. A stacking ensemble for network intrusion detection using heterogeneous datasets. Secur. Commun. Netw. 2020, 2020, 4586875.
13. Dutta, V.; Choraś, M.; Pawlicki, M.; Kozik, R. A deep learning ensemble for network anomaly and cyber-attack detection. Sensors 2020, 20, 4583.
14. Sahu, A.K.; Sharma, S.; Tanveer, M.; Raja, R. Internet of Things attack detection using hybrid Deep Learning Model. Comput. Commun. 2021, 176, 146–154.
15. Booij, T.M.; Chiscop, I.; Meeuwissen, E.; Moustafa, N.; den Hartog, F.T. ToN_IoT: The Role of Heterogeneity and the Need for Standardization of Features and Attack Types in IoT Network Intrusion Datasets. IEEE Internet Things J. 2021, 9, 485–496.
16. Alsaedi, A.; Moustafa, N.; Tari, Z.; Mahmood, A.; Anwar, A. TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. IEEE Access 2020, 8, 165130–165150.
17. Alghamdi, R.; Bellaiche, M. A Deep Intrusion Detection System in Lambda Architecture Based on Edge Cloud Computing for IoT. In Proceedings of the 2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 28–31 May 2021; pp. 561–566.
18. Zaharia, M.; Xin, R.S.; Wendell, P.; Das, T.; Armbrust, M.; Dave, A.; Meng, X.; Rosen, J.; Venkataraman, S.; Franklin, M.J.; et al. Apache spark: A unified engine for big data processing. Commun. ACM 2016, 59, 56–65.
19. Han, J.; Haihong, E.; Le, G.; Du, J. Survey on NoSQL database. In Proceedings of the 2011 6th International Conference on Pervasive Computing and Applications, Port Elizabeth, South Africa, 26–28 October 2011; pp. 363–366.

20. Yang, J.; Qu, J.; Mi, Q.; Li, Q. A CNN-LSTM model for tailings dam risk prediction. IEEE Access 2020, 8, 206491–206502.

21. Agatonovic-Kustrin, S.; Beresford, R. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. J. Pharm. Biomed. Anal. 2000, 22, 717–727.

22. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. Proc. IEEE 1998, 86, 2278–2324.

23. Lan, T.; Hu, H.; Jiang, C.; Yang, G.; Zhao, Z. A comparative study of decision tree, random forest, and convolutional neural network for spread-F identification. Adv. Space Res. 2020, 65, 2052–2061.

24. Aldweesh, A.; Derhab, A.; Emam, A.Z. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. Knowl.-Based Syst. 2020, 189, 105124.

25. Belgiu, M.; Drăguţ, L. Random forest in remote sensing: A review of applications and future directions. ISPRS J. Photogramm. Remote Sens. 2016, 114, 24–31.

26. Schapire, R.E. Explaining adaboost. In Empirical Inference; Springer: Berlin/Heidelberg, Germany, 2013; pp. 37–52.

27. Zhou, H.; Zhang, J.; Zhou, Y.; Guo, X.; Ma, Y. A feature selection algorithm of decision tree based on feature weight. Expert Syst. Appl. 2021, 164, 113842.

28. Boualouache, A.; Engel, T. A Survey on Machine Learning-based Misbehavior Detection Systems for 5G and Beyond Vehicular Networks. arXiv 2022, arXiv:2201.10500.

29. Rish, I. An empirical study of the naive Bayes classifier. In IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence; IBM: New York, NY, USA, 2001; Volume 3, pp. 41–46.

30. Fan, J.; Ma, X.; Wu, L.; Zhang, F.; Yu, X.; Zeng, W. Light Gradient Boosting Machine: An efficient soft computing model for estimating daily reference evapotranspiration with local and external meteorological data. Agric. Water Manag. 2019, 225, 105758.

31. Hancock, J.T.; Khoshgoftaar, T.M. CatBoost for big data: An interdisciplinary review. J. Big Data 2020, 7, 1–45.

32. Saha, A.; Subramanya, A.; Pirsiavash, H. Hidden trigger backdoor attacks. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20), New York, NY, USA, 7–12 February 2020; Volume 34, pp. 11957–11965.

33. Mukhopadhayay, I.; Polle, S.; Naskar, P. Simulation of denial of service (DoS) attack using matlab and xilinx. IOSR J. Comput. Eng. (IOSR-JCE) 2014, 16, 119–125.

34. Ali, O.; Cotae, P. Towards DoS/DDoS attack detection using artificial neural networks. In Proceedings of the 2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile

Communication Conference (UEMCON), New York, NY, USA, 8–10 November 2018; pp. 229–234.

35. Alenezi, M.; Nadeem, M.; Asif, R. SQL injection attacks countermeasures assessments. Indones. J. Electr. Eng. Comput. Sci. 2021, 21, 1121–1131.
36. Conti, M.; Dragoni, N.; Lesyk, V. A survey of man in the middle attacks. IEEE Commun. Surv. Tutorials 2016, 18, 2027–2051.
37. Subangan, S.; Senthooran, V. Secure authentication mechanism for resistance to password attacks. In Proceedings of the 2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, Sri Lanka, 2–5 September 2019; Volume 250, pp. 1–7.
38. Maniath, S.; Ashok, A.; Poornachandran, P.; Sujadevi, V.; AU, P.S.; Jan, S. Deep learning LSTM based ransomware detection. In Proceedings of the 2017 Recent Developments in Control, Automation & Power Engineering (RDCAPE), Noida, India, 26–27 October 2017; pp. 442–446.
39. Phase, E. Scanning and Enumeration Phase. In Constructing an Ethical Hacking Knowledge Base for Threat Awareness and Prevention; IGI Global: Hershey, PA, USA, 2019.
40. Fang, Y.; Li, Y.; Liu, L.; Huang, C. DeepXSS: Cross site scripting detection based on deep learning. In Proceedings of the 2018 International Conference on Computing and Artificial Intelligence; Association for Computing Machinery: New York, NY, USA, 2018; pp. 47–51.