electronics

MDPI

Article

Applications in Security and Evasions in Machine Learning: A Survey

Ramani Sagar [1,*], Rutvij Jhaveri [2] and Carlos Borrego [3]

[1] Computer/IT Engineering Department, Gujarat Technological University, Ahmedabad 382424, India
[2] Department of Computer Science & Engineering, Pandit Deendayal Petroleum University, Gandhinagar 382007, India; rutvij.jhaveri@sot.pdpu.ac.in
[3] Department of Information and Communications Engineering, Autonomous University of Barcelona, 08193 Barcelona, Spain; carlos.borrego@uab.cat
* Correspondence: sagarramani@gmail.com; Tel.: +91-9408109793

Received: 4 October 2019; Accepted: 11 December 2019; Published: 3 January 2020

check for updates

Abstract: In recent years, machine learning (ML) has become an important part to yield security and privacy in various applications. ML is used to address serious issues such as real-time attack detection, data leakage vulnerability assessments and many more. ML extensively supports the demanding requirements of the current scenario of security and privacy across a range of areas such as real-time decision-making, big data processing, reduced cycle time for learning, cost-efficiency and error-free processing. Therefore, in this paper, we review the state of the art approaches where ML is applicable more effectively to fulfill current real-world requirements in security. We examine different security applications' perspectives where ML models play an essential role and compare, with different possible dimensions, their accuracy results. By analyzing ML algorithms in security application it provides a blueprint for an interdisciplinary research area. Even with the use of current sophisticated technology and tools, attackers can evade the ML models by committing adversarial attacks. Therefore, requirements rise to assess the vulnerability in the ML models to cope up with the adversarial attacks at the time of development. Accordingly, as a supplement to this point, we also analyze the different types of adversarial attacks on the ML models. To give proper visualization of security properties, we have represented the threat model and defense strategies against adversarial attack methods. Moreover, we illustrate the adversarial attacks based on the attackers' knowledge about the model and addressed the point of the model at which possible attacks may be committed. Finally, we also investigate different types of properties of the adversarial attacks.

Keywords: security; privacy; adversarial attack; machine learning; attackers' knowledge

1. Introduction

The present-day community accesses advanced technologies, both hardware, and software, at an unprecedented pace in possibly every imaginable field. However, this has resulted in a whole new range of threats in terms of privacy and security. Therefore, there is a demanding need to address the security and privacy perspective of different types of cyber threats which are increasing at a drastic pace with unknown malware [1]. According to a special report [2], out of seven billion population in the world, about six billion rely on mobile phones or other smart gadgets for banking, shopping, financing, healthcare, internet-of-things (IoT), blockchain applications, posts on social media and for professional information and updates [3–7]. Therefore, during downloading of the applications on smart devices, there is a strong chance of data leakage and theft. Apart from that, malware is also triggered by corrupt system routines, unauthorized network access to resources and gather sensitive information. To cope up with these issues, many anti-virus tools, intrusion detection systems [8], defenders, and latest firewalls with updated security patches are available. However, according to the aforementioned report [9], malware distribution continues to grow at more than 267% per annum worldwide.

From the security perspective, the core research is focused on (1) dynamic vulnerability analysis; (2) static vulnerability analysis and; (3) hybrid vulnerability analysis. Even though static vulnerability analysis techniques have agility, it generates a high false-positive rate which shows less accuracy [10]. Meanwhile, dynamic vulnerability analysis techniques are accurate, but only for the substantial system. At the same time, accuracy gets compromised while adopting these techniques. Hybrid techniques attempt to overcome both these issues addressed in static and dynamic techniques. However, hybrid techniques are able to detect new types of vulnerabilities [11]. In recent times, hardware and software vendors have introduced many new techniques such as data execution protection, space layouts randomization, structured exception handler overwriting protection [12] and mandatory integrity control [13]. We claim that current evasion techniques can be easily bypassed and vendors are still in a developing phase in order to handle severe sophisticated attacks.

Recently some surveys on security applications in the context of machine learning and artificial intelligence have been presented [8] ML techniques for cybersecurity with an emphasis on ML methods and their description. Many other papers represented these methods have been published including many reviews. Also, previous works either focus on adversarial techniques or defense techniques of the machine learning classifiers. While this paper target work comparison of security applications as well as adversarial aspects including its defense techniques also during every phase of the machine learning life cycle from a data-driven view. The fundamental difference between previous surveys which have been proposed by authors, most of them only involve only security threats, internal issues of the machine learning systems in terms of adversarial defense. While in this survey based on that circumstance this survey combines different security applications and studies and carries out comprehensive summery in terms of tables based on the various parameters. Also, this survey highlights adversarial attack properties and attacks defense techniques for security applications in which ML plays an essential role. We emphasize a detailed review of security application with its performance matrices comparison as well as data distribution drifting leads by adversarial samples and private information transgression problem and its defense with attack model. This survey, as a complete summary combines numerous references and provides a macro understanding and interrelationship of security applications and machine learning related fields. This paper is intended for readers who wish to begin research towards the field of security application using ML techniques. As such great emphasis is placed on the thorough description is given about security application as well as the adversarial setting during the ML lifecycle.

With the explosion of data accelerating at an exponential rate, the privacy of data and systems has also come into the foreground. Privacy can be seen as one with a wide scope under a big umbrella [14]. In applications such as banking, healthcare, and defense, not only the issues related to privacy is a serious concern, but also there are legal concerns that need to be considered as mentioned in the Health Insurance Portability and Accountability (HIPAA) act [15]. Within the current technological scenario, there is a well-known discipline called privacy-preserving' [13] and statistical disclosure control (SDC) [16]. When data is distributed, preserving privacy becomes more challenging. Some of the statistical disclosure control techniques which address risks related to data disclosure [17] and privacy-preserving techniques in the frame of signature-based detection are not really foolproof [18]. In order to preserve data privacy, it becomes imperative to address the issues concerning the balance between false-negative rate, false-positive rate, recall, precision and performance [19].

The remainder of the paper is structured as follows: Section 2 states survey methodology which describes the taxonomy of the security applications and based on that how survey criteria forms. Section 3 reviews and analyzes different types of security applications for which ML approaches can be applied in order to prevent security threats on the applications. Apart from this, Section 3 gives complete information about all types of performance metrics used in the approaches of IDS with the type of different type classifiers with algorithms, limitations and future challenges. Section 4 presents a vulnerability analysis threat model different types of adversarial attacks at each level on the ML classifiers from different types of attackers. Also, we review adversarial defense techniques for the different types of adversarial attacks that cover both reactive and proactive types of defense techniques.

2. Survey Methodology for Security Applications

The idea behind conducting a survey is deceptively simple which involves identifying different types of application where machine learning classifier is involved in the applications. This section introduces a taxonomy of different security applications where machine learning can be applied in order to fulfill the desired goal. Several survey machines learning-based survey papers have been proposed but most of them address security-based issues in machine learning applications. Therefore, considering the above circumstances, this survey in addition to that also includes privacy with security aspects and adversarial attacks on machine learning classifiers.

To design the survey for the security-based application we divide the security spectrum comprehensively into different applications which include the comprehensive method analysis for the application in all aspects of security. Also, we expand the spectrum in terms of the type of classifiers where it requires analyzing especially for intrusion detection as well as prevention. Although the scope of the security applications is broad which can not be limited for few applications in the recent technological advancement. In this survey, we have considered security applications in which machine learning plays a vital and essential role in security applications. Because ML algorithms comprise and designated for statistical mechanisms such as decision tree, logistic regression, and function approximations. This type of algorithm is more influential and can be used in that type of situation where classification is essential. The machine learning technique imposes many advantages especially when it applied in security aspects like (1) signature-based attack where subtle changes in the signature can be discovered dynamically, (2) system behavior and identify anomaly from the deviation of the normal system, (3) by stabilizing biased variance, recall verses precision machine learning provides lower sensitivity and reduces false alarm rate, (4) machine learning is highly recommended when the domain of the threat model is changed.

2.1. Study Selections and Search Methods

The study selection process for the survey includes mainly three phases: (1) title and abstract review, (2) classification of security applications, (3) review of security properties and defense techniques. Inclusion criteria for a title and abstract review are security applications where machine learning methods are applicable, weakness of ML models where attackers craft the attacks and identify the methods to defend the attacks. In the second phase of the review process, selection criteria of the survey described are base on machine learning in security applications. Based on the inclusion criteria total of one hundred fourteen papers satisfy the inclusion criteria and classified according to presume array of variables. In the third phase of the review process selection criteria of the survey describe based on "Adversarial attacks on machine learning and defense". Based on these inclusion criteria total of twenty-three papers satisfies the inclusion criteria.

For the survey, bibliographic databases are explored using an online search interface for paper selection. To achieve precise relevancy to the title in Table 1 we have described the query format for searching through interfaces.

Table 1. Search queries format for search engine.

Search Engine	Query
IEEE Xplore	(machine learning AND applications AND security) OR (Adversarial Attacks AND machine learning) [1]
Springer	(machine learning AND applications AND security) OR (security properties OR defense AND machine learning) OR (adversarial attacks AND machine learning) [2]
Web of Science	(machine learning AND application AND security) OR (security properties OR defense AND machine learning) OR (adversarial attacks AND machine learning) [2]
Science Direct	(machine learning AND application AND security) OR (security properties OR defense AND machine learning) OR (adversarial attacks AND machine learning) [2]

[1] Searched in Metadata; [2] Search topics filtered by subjects: Computer Science; Security Applications; Privacy; Machine Learning; Adversarial attacks.

2.2. Variable Definitions

In this review, variables are considered based on general characterization, security properties, and adversarial analysis. Table 2 characterizes detail criteria regarding study characterization, type of algorithm used for modeling, type of metric used for result analysis, consideration of data acquisition techniques for classifiers training and testing, and type of application approach. While, as per the adversarial point of view, detailed criteria regarding types of security property consists of an existing approach and defense strategy used for the approach. In Table 2 all criteria are represented in the respective sub-sections.

Table 2. Types of variables considered for paper selection during the review.

Variable	Explanation
Study	
Year	Publication year
Type of Study	Rationale regarding outcome analysis
Problems	Rationale of problem identification, address, and solution
Security applications	
Methodology	Types of machine learning algorithms used in the approach
Type of classifier	Which type of classifiers are adopted for simulation
Performance metrics	Types of performance metrics and accuracy achieved
Data acquisition	Type of datasets are used in training and testing
Application approach	wether security applications anomaly, misuse or hybrid detections based approach used
Adversarial analysis	
Security properties	Type of security property consists of the existing approach
Defense approach	Which type of defense approach

2.3. Time Period Considered

In order to analyze the research trends of machine learning in security applications and adversarial attacks, we have divided the time period into five different intervals. We have considered the time period range first interval from 2016 to 2019, for the second interval from 2011 to 2015, for the third interval from 2006 to 2010, for the fourth interval from 2001 to 2005 and for the fifth interval from 2000 and below.

2.4. Studies Characterization

This subsection represents the statistical results retrieved from the search methods and types of variables considered in the variable definition criteria. Figure 1 illustrates the distribution of published papers for the last 25 years. In the graph, the time period of total years is represented in five different intervals. The result represents an increasing trend of machine learning methods in security applications and adversarial attack techniques. It is observed in Figure 1 where activity in the machine learning domain is increased since 2006. Apart from this in the period 2016–2019 flourishing trend of the published papers in the area of machine learning-based security applications and adversarial attacks.

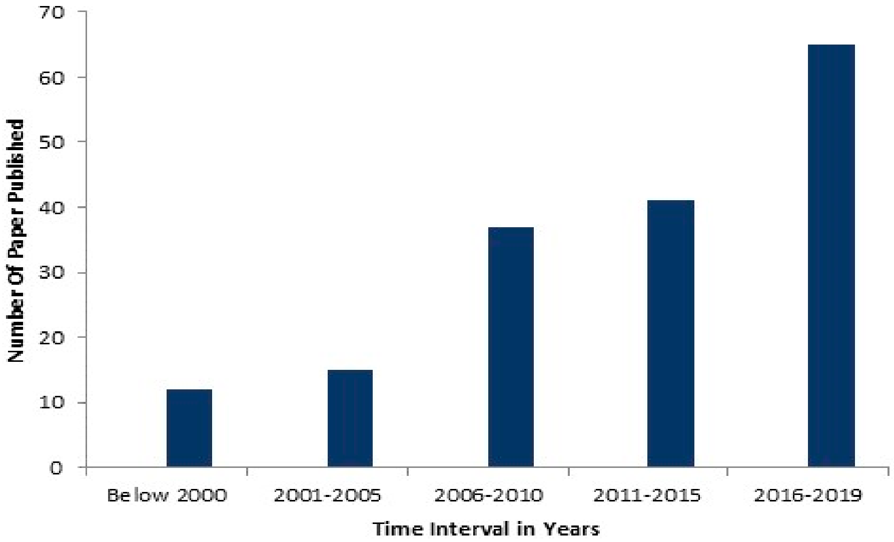

Figure 1. Time interval distribution of published papers focusing based on.the use of machine learning in security applications and adversarial attacks.

The taxonomy of security applications where machine learning is applied is illustrated in Figure 2.

- Intrusion can be separately classified into intrusion detection and intrusion prevention techniques. Further intrusion detection can be classified into an anomaly-based and signature-based approach.

    1. Anomaly-based intrusion detection detects misuse in a computer or network with the help of machine learning classifiers either normal or anomalous.
    2. While signature-based detection is identified by the ML classifiers algorithms by identifying specific patterns such as malicious instruction sequences or byte sequences.

- Intrusion prevention is a preemptive approach that identifies potential threats with the help of ML classifiers and responds to them accordingly in order to prevent misuse.
- Phishing detection intended to detect legitimate or phishing web pages and applications which mainly exploit computer users' vulnerability with the use of ML classifiers.
- Privacy preservation is another important aspect of security where in order to provide security of sensitive information during communication between different parties. Here ML classifiers help to prevent leakage of the sensitive data with other collaborative entities. Linear Means classifier

simply computes a multiple of the original LM score function with the same sign and algorithm made confidential by encoding all real vector coefficients as integers and encrypts the input vectors coefficient wise and carries out the linear algebra operations with vectors of ciphertexts. While Fisher's Linear Discriminant Classifier same procedure is done like LM classifiers but using gradient descent using different weight vector.

- Spam detections like blatant blocking, bulk email filter, category filter, null sender header tag validation, and null sender disposition ML classifiers are used.

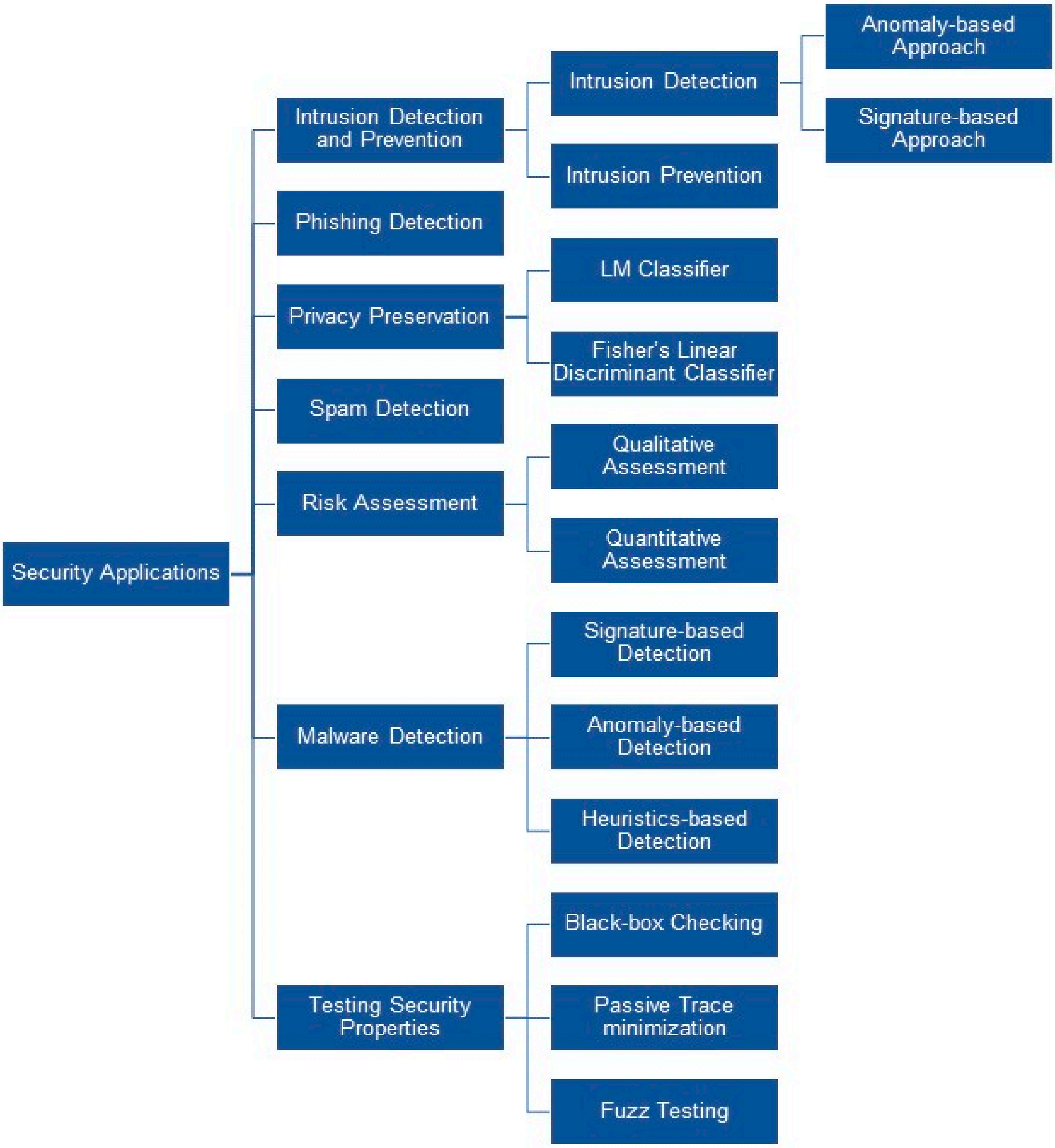

Figure 2. Taxonomy of Security Applications.

Risk assessment is also an important factor which classifies different information assets that possibly damaged and identify various risk that could affect the assets with the help of ML classifiers.

- Malware detection, ML classifier formalizes and finds the principals that inhibit the data it examines. If a previously unseen sample is found then it could be the new file and based on the properties it contains the decision has been taken about malware detection. Based on the type of the signature we categorize malware detection techniques into signature-based detection, anomaly-based detection, heuristics based detection.
- Testing Security properties require to ensure the safety and authenticity of the protocol systems. In order to model the testing process in an automated manner, using three techniques namely black-box checking, passive trace minimization, and fuzz testing to fulfill the desired goal of testing security.

3. Machine Learning Applications in Security

3.1. Intrusion Detection and Prevention

In recent years, Internet service has played an important part in business models. Since both the customer and the business use Internet applications, the security of data while utilizing the Internet as a medium has become a prime concern [20]. The intrusion detection system (IDS) provides a defense to counter the attacks [21,22]. In IDS, several approaches are proposed such as packet payload analysis [23], pattern propagation [24] and bro language [25]. In addition, various schemes for ad-hoc networks are proposed to detect attack patterns and to provide a defense mechanism in the network [26–30]. IDS which are passive by nature have a key issue their inability to mount targeted, reliable and adaptive response [31]. Therefore, sometimes host-level IDS does not assure how a packet is processed which may lead to wrong decisions [32] and that is why the adaptive and proactive system is required over IDS. In [33], the authors state anomaly-based and signature-based approaches for IDS as follows.

3.1.1. Intrusion Detection System Approaches

In this Segment, we review types of approaches for IDS. In the IDS, essential factors cause of error in classifier learning due to noise, bias, and variance. Therefore, ensemble and hybrid classifiers are the types of multiclass classifiers that help to minimize these factors due to the bagging, boosting and stacking properties in the classifier models. Based on these criteria we have analyzed a signature-based and anomaly-based approach.

Signature-Based Approach

A signature-based intrusion detection technique uses a pre-defined pattern to identify malicious activity [21] while in the traditional methods, it may not be able to intimate the system about new threats.

1. Single classifier

In this type of intrusion detection technique, only a single ML algorithm is used to detect the intrusion. Akira et al. [34] proposed a decision tree algorithm with the Gini index, which engenders refined data that are used to learn the classifiers for raising the alerts as an output of signature-based IDS. Lippmann et al. [35] represent a theory that detects the signature of a known attack by examining attack-oriented basic keywords in the network. Network sniffing data are used to produce the count of the keywords in each telnet session. Counting of each keyword occurrences is used for detection by the neural network classifier. Wong et al. [36] illustrated an artificial neural network (ANN) as well as a support vector-based classifier approach to predict the types of attacks, which are based on frequency-based encoding techniques. ANN is trained with the backpropagation algorithm to predict the intrusion. Additionally, a support vector machine (SVM) model is also built in order to classify attack. From the observation of both techniques, it shows that SVM gives better results than ANN for the same encoding method of detection.

2. Hybrid classifier

This type of classifier primarily a blend of the heterogeneous environment or/and classifiers as a detection mechanism in which right from the data normalization phase to the final decision phase is covered. Borges et al. [37] presented a communication control module, a monitoring module, a mobile correlator module and, command and control center (C&C) components described to which the command and control center (C&C) center consists of a hybrid classifier. The monitoring module is responsible for monitoring normal and abnormal patterns for file access and usage, content observer and a broadcast receiver. All the information thus gathered is used afterward by the ML classifiers to access if any user or mobile device's application face any security threats. Karthick et al. [38] presented a two-stage framework: the first stage naïve Bayes classifier is employed to raise a flag that identifies malicious activities in the network and incoming traffic fed as an input to the hidden Markov model (HMM). HMM is an effective approach to blacklist IP addresses based on suspicious characteristics of the traffic. The plot model [36] proposed by Tsai et al. [35] (triangle area-based nearest neighbors) uses K-means clustering to conclude the cluster center corresponding to the attack classes. Two cluster centers and one data among the datasets are used to calculate the area of the triangle and form a signature from that K-Nearest Neighbour classifier which is employed to identify threats.

3. Ensemble Classifier

The ensemble is a combined classifier of multiple weak classifiers. In this method, weak learners are trained so that the inclusive action of the model can be adequately alleviated. For improving the performance of the weak learner, adaptive boosting [38], bagging [39], wagging [40], random forest [41] and cross validators committees [42] play a crucial role. Ma et al. [43] proposed a scheme that combines the deep neural network (DNN) and spectral clustering algorithms. Datasets are calved into the K subsets using cluster centers. Based on the similarity features, distance is measured among the data points in training sets and testing sets which are employed into the model of DNN to detect the intrusion. In Table 3, we analyze different ML approaches for signature-based intrusion detection applications for all three types of classifiers. Our criteria for Table 3 for signature-based intrusion detection system based on which type of classifiers are used in the approach from above mentioned three types of classifiers that is single, hybrid or ensemble classifier.

Table 3. Analysis of Signature-based Intrusion Detection Techniques (Notations: SVM-Support Vector Machines, TANN-Triangle Area Based Nearest Neighbor).

Reference	[34]	[35]	[36]	[37]	[38]	[43]	[44]	[45]
Type Of Algorithm	Decision tree algorithm	Neural network	Artificial Neural network, SVM	k-means clusturing + semi-supervised Algorithm	Hidden Markow Model + Naïve Biase	Spectral clustering + Deep Neural Network, SVM, Random Forest, Back Propagation Neural Network	Genetic Algorithm	TANN, SVM, k-means + k-NN
Advantage	No data preparation required	Ability to detect a new type of attack with less computation	Frequency-base encoding is more effective for intrusion detection	Low impact on mobile in terms of battery consumption, CPU/memory usage	Highly Accurate	Accuracy is high for attack detection, error rate is low for all tested data sets, able to classify sparse attacks	Adopts changes in environment, robust against noise and detects unknown types of attacks	Accuracy rate and detection rate is high for single class
Limitation and challenge	False positives found for known attacks in defense advance research project agency	Stealthy attack which hides the keywords requires additional action	Improper parameter selection cause overfitting in ANN	Selective policy implemented for monitoring suspicious events	HMM requires more than 5 states to maintain high accuracy	Requires to optimize weight parameter and threshold for each DNN layer	Compared to rule optimization method false positive rate is high	Low accuracy for more than one class
Performance Metrics	True positives, false positives, false negatives, true negatives	False alarms Detection Rate—80%	Attack detection Rate—95.9%, false positive rate	CPU usage, RAM usage, battery usage, sent/Received packets	Accuracy—100%, false alarm rate	Accuracy—91.97%, recall—92.23%, error rate—7.9%	False positives and intrusion detection rate—99.6%	Accuracy—96.91%, false positive rate, detFalse positives and intrusion detection rate—99.6% ection rate—98.95%
Classifier type	Single	Single	Single	Hybrid	Hybrid	Ensemble	Single	Hybrid

Anomaly-Based Approach

This type of technique observes the system behavior and identifies anomaly from the deviation of the normal system. Hence, this type of system has the ability to detect the zero-day attack [32]. Using this technique normal behavior of the system can be customized so that, for the adversary, it is difficult to figure out the normal behavior of the system. In Table 4, we investigated anomaly-based IDS approaches based on the types of classifiers.

1. Single classifier

Van et al. [46] proposed a deep learning technique to implement an anomaly-based network IDS (NIDS). In the proposed model, the deep belief network structure is constructed that consists of components such as stacked auto-encoder and stacked restricted Boltzmann machines (RBM). With the use of auto-encoder, the classifier learns and extracts hierarchical features by unsupervised or semi-supervised learning. The RBM probabilistic model aims to use the natural criteria to maximize the log-likelihood. Therefore, the proposed method can be enabled to detect attacks and classify them into these classes: probe, user to root, normal, remote to user and denial-of-service (DoS). In [47], the authors presented a scheme for creating intrusion database with the main objective to create an easy-to-update database tool which simultaneously produces real traffic data. To make the ML scheme effective, the proposed method is used as a multi-objective feature selection technique that acknowledges substantial network characteristics that yield higher accuracy. Ghanem et al. [48] opined that anomaly-based IDS aims to accomplish an excessive number of false alarms. They constructed an SVM based-machine learning technique that complements the performance of the IDS and also decreases the false alarm rate vigorously. In the performance assessment projected, the unsupervised IDS approach detects all malicious traffic and reduces false alarms compared to one-class and two-class linear and non-linear SVM approaches.

2. Hybrid classifier

Al-Yaseen et al. [49] illustrated a modified k-means algorithm that aims to achieve high-performance and considers all possible eventualities by treating all the divergent points in the datasets as the initial centroid of the cluster instead of selecting a specific set of initial centroid randomly. In addition to this, the modified k-means clustering standard C4.5 technique [50] builds a tree from the clusters which can detect the anomaly by using the maximum information gained from the feature selection. Consequently, minimum information is split to build a tree structure of normal and malicious behaviors. Abadeh et al. [51] proposed a parallel genetic local search algorithm that is capable of generating fuzzy rules for detecting intrusive behavior in the networks. In this approach, the population is divided into subpopulations, which are the number of classes for the classification analysis or problem, and training set for each classifier is different from each other. The fuzzy ruleset evolves independently in a parallel manner and is used as a source of knowledge for each classifier for intrusion detection.

Table 4. Analysis of Anomaly-based Intrusion Detection Techniques.

Reference	[46]	[47]	[48]	[49]	[51]	[52]	[53]
Type of Algorithm/Technique	Deep learning	Decision tree, naïve Bayes	Support vector machine	Hybrid modified k-means + C 4.5, Bayes net, Naïve Bayes, LibSVM, JRip, sequential minimal optimization, k Instance base	Genetic local search algorithm	Parzen density estimation, k-means, v-SVC (Support vector classifier)	Bagged tree, ada boost, RUS boost, logit boost, gentle boost
Advantage	Accuracy of intrusion detection and attack classification is high	Considered real-world network properties to increase accuracy	Datasets comprise with non-homogeneous features, shows high accuracy	Construct clusters with all cases characterized by significant differences among the instance which leads to higher accuracy	Needs less training time, high detection rate, low false alarm rate	v-SVC algorithm performs better than its competitors, high detection rate, low false alarm rate, accurately models the traffic of each service	Provides comparative study of ensemble ML methods for imbalance datasets
Limitation and challenge	Training for stack auto-encoder is time-consuming	False-positive rate increases while detecting new types of attacks	IDS accuracy decreases when analyzing the probing datasets	Precision is less compared to other methods like naïve Bayes, J48 NB tree	Discovered trade-off between false alarm and detection rate with the value of local search procedure	False alarm rate is increased when the detection rate is increased for the approach	Needs to require designing ML algorithm variations with all relevant security issues
Performance Metrics	Accuracy—99.98%, error rate—0.02%	False positives, false negatives, accuracy—97.42%	False-negative rate—0%, detection rate—100%, false positive rate—0%, overall success rate—100%	Accuracy—91.13%, detection rate—85.26%, false alarm rate—2.99%, precision—96.65%, specificity—97.01%, F-measure—90.56%	Detection rate—92.4%, classification rate—93%, false alarm rate—0.15%	Detection rate—94.31%, false alarm rate—9.49%	Sensitivity, specificity, precision—99.3%,
Type of classifier	Single	Single	Single	Hybrid	Hybrid	Ensemble	Ensemble

3. Ensemble Classifier

Giacinto et al. [52] proposed an unlabelled network anomaly IDS based on the modular multiple-classifier systems. The aim of this modularized design to develop a batch of homogeneous network or protocol services that grant the designer to choose decision thresholds and different models for the module to enhance the comprehensive performance of the ensemble model. For ensemble classifiers, the posterior probability function is used on which min, max, mean and product rule can be easily applied and, traditional classifiers can be combined using the rules for obtaining the detection result. While [53] uses AdaBoost, RUSBoost, LogiBoost, and gentle boost Bagged tree algorithm for ensemble classifiers and results are compared with each of the referred algorithms. As a result of these algorithms, bagged tree and gentle boost classifiers show notable performance. In Table 4, we classify anomaly-based intrusion detection techniques for the aforementioned different types of classifiers.

3.1.2. Intrusion Prevention

Intrusion prevention is a vulnerability prevention technique that monitors network flow to detect and prevent dicey traffic misuse. Intrusion prevention is an extension of IDS however, both investigate malicious activity in network traffic. One critical difference in intrusion prevention over intrusion detection is that intrusion prevention is to construct and design more active protection to enhance intrusion detection. This type of approach is most suitable where it is essential to react in realtime to prevent or block malicious activities. In [54], the authors addressed the issue of cyber terrorism and emphasized that the response and defense mechanism of any system must be robust, adaptive and efficient. They proposed a genetic programming mechanism for the prohibition of cyber-crime. The basic weapons of cyber-terrorists are a modified version of intrusion methods such as spoofing, email bombs, data sniffing, parasites, worms, backdoors, DoS attacks [55], Trojan horse [56]. In the proposed work, the authors used a genetic algorithm to investigate the issues with the use of Knowledge-Data Discovery (KDD)-Cup dataset which was provided by the Massachusetts Institute of Technology (MIT) laboratory and Defense Advanced Research Projects Agency (DARPA). With the use of homologous crossover and machine coded linear genomes operator, the desired result has been found and they have been able to detect and predict the malicious intrusions. The proposed hybrid approach of feedback intrusion prevention system in [52] protects binary code injection attacks. It contains three main units: (1) signature-based filtering scheme; (2) anomaly-based classifier and; (3) supervision framework. The supervision framework utilizes instruction set randomization, which prevents the code injection attacks and identifies malicious code, which can be used to learn the classifiers and filtering scheme as feedback. Consequently, it is capable enough to refuse the zero-day attack or metamorphic types of attacks by nature.

3.2. Phishing Detection

Phishing is a technique to steal personal and sensitive information of the victim by enticing the users to visit a fake email or web pages to mimic the victim's own page visual identity. Phishing attacks cause damage to a victim's personal and sensitive information by spoofing email [57], fake social network accounts [58] and hacking [59]. To detect phishing attacks, many approaches have been proposed such as DNS-based blacklist, automated individual whitelist, heuristic, and visual similarity and, ML-based techniques. Out of all these, MLbased techniques can automatically detect zero-hour attacks effectively on a large scale basis [60].

Xiao et al. [61] represent phishing attacks built on the semisupervised ML approach which is implemented on a transductive support vector machine. For the feature extraction of the web page, they used the document object model which also includes a gray histogram, color histogram and spatial relationship among the subgraph to leverage the phishing detection including some web image features. TSVM takes into consideration how the distribution information implicitly exhibits large quantity unlabeled datasets that provide effective performance as compared to SVM. In [62], it was reported that solutions for discovering phishing attacks in the online mode, which are based on state-of-art techniques, suffer from lack of accuracy. Therefore, authors addressed this issue with the help of a neural network with reinforcement learning. In the preprocessing phase, features were selected from each header, email text, URL and HTML content which are given as an input to feature evaluation and reduction algorithm (FERA). This aims to decide a number of defense features to be applied to the classification process and also accelerates the adaption process in the neural network. The DNN is analyzed and online emails are classified. A reinforcement learning agent is used to acknowledge diversity in the online datasets and to provide a decision about legitimate or phishing emails. The proposed approach is shown to have a false positive rate of 1.8% and an accuracy of 98.6%.

Hamid et al. [63] formulated an approach for clustering techniques and email phishing detection profiling. In the feature selection, information gain is used to give weight to the attribute for a set of training feature vectors. For the profiling, a two-step clustering algorithm is developed to deal with a large set of data for handling continuous and categorical data. In the primary stage, profiles are generated based on the clustering algorithms' prediction and therefore, a cluster represents the profile of elements in the prediction of phishing emails. In the second stage, profiles are employed to train the classification algorithm for predicting the unrecognized class labels from the input data. For emails categorization, AdaBoost and sequential minimal optimization algorithm are used which are ensemble type classifiers and, their performances vary different datasets and number of clusters.

Basnet et al. [64] discussed that the basic behavior of adversaries can be obtained through email headers. Email filtering is divided into two types: (1) content-based filtering and; (2) origin-based filtering. For feature selection, the authors used the wrapper-based feature selection method and the correlation-based feature selection method. For searching into the feature subset within the time constraint, as a result, they have presented a greedy forward search and genetic algorithm. They showed that the wrapper-based feature selection method is slower related to correlation-based feature selection while wrapper-based feature selection methods have better accuracy over the classifiers as compared to the correlation feature selection method. The mentioned techniques with types of feature selection sets can be bundled up into the classifiers that are shown in Table 5.

Table 5. Analysis of phishing detection techniques (notations: feature extraction and reduction, CFS- correlation-based feature selection, WFS- Wrapper Feature Selection).

References	[63]	[64]	[65]	[66]	[67]
No. of Feature Sets	20	177	5	50	7
Feature Selection Method	Information gain	CFS, WFS	Segmentation	URL segmetation	Hybrid
Type of Algorithm/Technique	AdaBoost, sequential minimization, optimization	Naïve Bayes, logistic regression, random forest	Support vector machine, transductive SVM (TSVM)	Neural network, reinforcement learning	Bayesian Network algorithm
Advantage	Low false positive and false negative rates, overfitting issues, solve as produced a less error rate	Using greedy forward search technique of selected feature sets yield higher accuracy and improved training time	Higher accuracy and higher precision compare to SVM	Able to recognize zero-day phishing attack	Able to achieve high accuracy for fusion of content-based and behavior-based approach
Limitation and challenge	Accuracy could be improved by integrating k-means and two-step clustering approach	Slower technique, degradation of classification accuracy for a new type of datasets	Less flexible in terms of learning	Does not mitigate various privacy concern and malicious bots	As feature selection is content-based, image content attack can bypass this approach
Performance Metrics	Error rate—18%, false positives, false negatives, accuracy—98.4%	Error rate—1.6%, false positive rate—1%, false negative rate—2.7%	Accuracy—91.1%, precision, recall	Precision—86.7%, recall—88%, accuracy—90%, F-measure—87.3%	False negatives—4.2%, false positives—4.1%, precision—96%, recall—96%, error—4%, accuracy—96%
Types of ML Classifiers	Ensemble	Hybrid	Single	Hybrid	Hybrid

3.3. Privacy Preservation

The prime objective of an ML technique is to extract the needful information from the data by its classifiers while preserving privacy by masking/hiding the sensitive data from the adversary [68]. Therefore, there is a need to balance these aspects while sensitive data is to be mined. To analyze the vulnerabilities in the privacy-preservation, several researchers have proposed attacking techniques such as minimal attacks, background knowledge attacks [69], additive data perturbation and homogeneity attacks. To combat the types of attacks, many approaches such as l-diversity [70], t-closeness [71], k-anonymity [72] and double-blinding [73] are represented by the researchers. Even with these types of approaches, one is not able to prevent the adversaries who already have knowledge about the datasets. As a result, it is imperative to protect data during the training phase in the ML techniques which are used for privacy-preserving.

Jia et al. [74] proposed a model that preserves privacy in ML for distributed systems. It is unreasonable for a distributed system to share the datasets between classifiers due to privacy concerns. The proposed approach prevents the leakage of private information from the learned model to other collaborative entities. This work also focuses on the confidentiality of the learning data before processing. To ensure that data classification is performed successfully without exposing to the tester, oblivious evaluation of multivariate polynomial approach is applied to the SVM classifier. In [75], the authors presented an ML algorithm in which estimations are considered as a function of the input data which can be proclaimed as polynomials of the bounded degree. Therefore, classification and training are carried out homomorphically on encrypted data. Table 6 reviews the performance of this scheme. With the quick evolution and wide-spread applications of cloud computing [76], research works have been published to build an outsourcing computation system over the cloud. However, how to securely outsource the computation to the cloud is a major challenge. Different solutions have been proposed which address this issue [77–80]. In the research of ML, Li et al. [76] discussed theoretical aspects of privacy in the sense of data privacy and privacy in the training model using deep learning. There is also a great need to focus on the computational cost of data owners to keep it to minimal. To preserve privacy when multiple actors engage in the deep learning model, before uploading data to the cloud, a multi-key fully homomorphic encryption (MK-FHE) scheme is proposed. Consequently, the authors proposed an advanced scheme, hybrid multi-key deep learning training system, which uses double decryption and fully homomorphic encryption (FHE). The training phase is executed over the ciphertext under a different public key. Therefore, a theoretical cloud model is able to train a deep learning model privately with additions and multi-functions which are semantically secure. Though the homomorphic cryptography has many important applications in operation for ciphertext and authentication [6,81,82], the efficiency is still a challenge for its practical applications. In [83], it is also proposed that to preserve privacy in the cloud, a deep computation model can be used for the purpose of big data feature learning. To protect sensitive data, the model uses Brakerski–Gentry–Vaikunathan scheme which provides encryption to the private data. This scheme is implemented to the cloud server in order to effectively enumerate the high order backpropagation algorithm to the encrypted data for the model training.

The sigmoid function is processed by the proposed technique as a polynomial function to help secure enumeration of the initiation function of the Brakerski–Gentry–Vaikunathan (BGV) scheme. The results of the illustrated method depict a 1% to 2% higher error rate and less accuracy as compared to the non-privacy preserving deep learning computational model. There are also some other related works that addressed the privacy protection issue in classification [84–87]. Because of the inefficiency problem with cryptographic solutions, there are also several related works presenting more efficient techniques including differential privacy [88–90].

Table 6. Performance comparison of the classifiers [75].

	Time for Classifying Test Vector	**Classifier Computing Time in the Training Stage**	**Time for Data Encryption in the Training Stage**
LM classifier (encrypted data)	Constant with number of training vector increases	Grows linearly	Grows linearly
Fisher's Linear Discriminant classifier (encrypted data)	Constant with number of training vector increases	Grows quadratically	Grows linearly

Zhang et al. [91] address a type of regularized observational risk minimization ML problem. The authors presented that alternate direction method of multiplier approach enables distributed training over the network and exchanges the result with its neighbors. During the exchange, the adversary can easily evade the sensitive data which can be protected by using dual variable perturbation and primal variable perturbation in order to guarantee the dynamic differential privacy. Furthermore, the authors showed the trade-off between accuracy and privacy. Accuracy is decreased as the privacy requirement increases further. In [92], the authors illustrated the use of the SVM privacy-preserving online medical pre-diagnosis framework. The model resides with the service provider with the main objective that when a query is received from the query engine, privacy and accuracy should not be compromised in response. To achieve this objective, the model uses non-linear SVM with lightweight polynomial aggregation techniques and multi-party random masking techniques. The computational and communicational overhead of the framework needs to be more efficient and suitable for medical pre-diagnosis services.

The authors in [93] proposed a privacy-preserving protocol using k-means clustering which provides cryptographic privacy protection for arbitrarily partitioned data. With the proposed method, two parties can be able to share their data in the arbitrary partition and able to learn k-means clustering of the shared data without exposing their data to each other. The output of the algorithm assigns 1 to k cluster numbers to each object. Both the parties learn the assignments if the bject is shared, otherwise, the assignment is given to the party to which the object belongs to. When the algorithm halts, the final mean of every cluster is shared by both the parties which then learn absolute final centers. Table 7 presents the review of privacy-preserving techniques and the type of sanitization methods aligned with the ML algorithm as a supplement to preserve privacy in an existing application.

Table 7. Review of Privacy Preserving techniques (Notations: OPME-Obvious Evaluation of Multivariate Polynomial, BGV-Brakerski-Gentry-Vaikunathan).

Reference	[74]	[75]	[76]	[82]	[91]	[92]	[93]
Application/Domain	Distributed system	Cryptography	Cloud Computing	Cloud and big data	Network	Healthcare	Data mining
Type of Algorithm	Support vector machine	Linear means classifier, Fisher’s Linear discriminant classifier	Deep learning	Deep learning	Altering the direction method of multipliers	Nonlinear Support vector machine	K-means Clustering
Advantage	Preserves privacy of data classification and similarity evaluation	Retain confidentiality of training and test data	Preserves privacy of data and training model	Efficiently deals with big data	Enables distributed training over network collaborative nodes and guarantees preservation of privacy at each update	Sensitive health information not disclosed during online prediagnosis services	Provides privacy preserving over horizontally, vertically and arbitrarily partitioned data
Limitation and challenge	increasing data, dimensions require more polynomials leads to high computational cost	Multiple data owners’ data cannot be handled using a single ML method without disclosing data	FHE scheme is practically not implemented	Gets excessive overhead to accomplish encryption/decryption and leads to deploying more cloud servers	Practical results of PVP and DVP are not as per a theoretical analysis	No of support vectors are fixed up to 60 in real environment experiments and overhead found for the computation	Intermediate cluster assignment potentially leak information
Performance Metrics	Classification accuracy- 97.21%, computational cost, data similarity	Accuracy, training time	Not Applicable	Training time, classification accuracy- 87.2%	Empirical loss, misclassification error rate, privacy-accuracy trade-off	Accuracy- 94%, computational complexity	Computational complexity
Sanitization methods as supplement to ML algorithm	OMPE	Homomorphic encryption scheme	Multi-key fully homomorphic encryption	BGV homomorphic encryption scheme	Dual variable perturbation, Primal variable perturbation	Polynomial aggregation techniques, light-weight multi-party random masking	Random shares, secure scalar product protocol, Yaos circuit evaluation protocol

3.4. Spam Detection

In the past few years, research interest has been increased for web-based services and systems, social networking and social media that incorporate large-scale data [94]. Several detection techniques have been proposed based on their classification. In [95], the authors proposed a technique to avoid spam distribution. Several algorithms, such as content-based spam detection [96], link-based spam detection [97], trust-based detection in IoT [98], real-time spam detection [99] and click spam detection [100] were proposed for spam detection. Spam detection systems have humongous data to be analyzed which involve multi-dimension attribute space with probably thousands of dimensions and is vigorous by nature [101]. ML grants tickets due to the adaptive capability to learn the patterns for classification of spam and no spam [102]. In Table 8, we review some machine learning-based spam detection techniques.

Chen et al. [103] represented ML streaming spam tweet detection technique which fills the gap between data feature and model by deriving performance evaluation. To collect the streaming tweets the author's used streaming application interface (APIs) with universal resource locator (URL) which provide 1% access to all public tweets and thus, collected 600 million tweets with the URL. To check whether the URL is malicious or not for labeling trend, Micro Web Reputation Services are used. For the feature set preparation, user-based and tweet-based features are extracted. As mentioned in Table 8, distinct types of ML algorithms are used for performance evaluation and analysis of the impact of increasing training data, the impact of different sampling methods, the investigation of time-related data in the form of detection rate and the average values of features.

Meanwhile, the authors in [104] presented a semi-supervised approach to detect the tweet spam which uses ensemble classifiers. In the framework, four classifiers perform the task with each having a different method to detect the spam. Classifier 1 uses a blacklist domain detector that checks URL with the use of questions filter. Classifier 2 analyzes the similarity of the tweet with the use of clusters and obtains spam and ham tweets. Classifier 3 uses a reliable ham tweet detector which conducts a content analysis of the posted tweets and text in it. Classifier 4 is preferred for increasing accuracy. The final labeling using classifier 4 is done as follows:

$$l_4^{tw_o} < -min \begin{cases} M^{KNN}(tw_o) \\ M^{NB}(tw_o) \\ M^{LR}(tw_o) \end{cases}, \quad (1)$$

where where $l_4^{tw_o}$ = final label of classifier 4, tw_o = tweet under observation, M^{KNN} = k-nearest neighbor model, M^{NB} = Naive Bayes model, M^{LR} = logistic regression.

At least two models should have the same label to identify a tweet as a ham or spam. After the final decision is taken using a majority voting approach, the database is updated in the framework. To make an update more adaptive and efficient, Markov bound-based update model is implemented. As per the results of the comparison between classifier1 to 4, classifier 2 has a higher precision rate and, classifier 4 has a higher recall rate and F1 score among all the classifiers.

Table 8. Review of existing spam detection. Techniques.

Reference	[103]	[104]	[105]	[106]	[107]
Type of ML Algorithm	Random Forest C4.5, Naïve Bayes, KNN Bayes Network, SVM	KNN, Naïve Bias, Logistic Regression	Neural Network	Multinomial Naïve Bayes, SVM, KNN, Random Forest Adaboost with a decision tree	SVM
Advantage	Enables real-time spam detection	Framework collects timely updates in an effective and adaptive manner	Detect deceptive review spam by linguistic and behavioral features	Reduces the overall error rate	Classifiers can be easily retrained as characters and tactics of sploggers changes
Limitation	Require continuous training and updating the data sets in order to maintain detection accuracy	As the framework is tested for tweeter only, other social media platform results may differ as features change	Proposed approach is compared with off-the-shelf classification algorithms	Due to the lack of real database of SMS a chance of under performance during the classification	Continuous updating of the feature sets is required
Performance Metrics	Precision, recall, F-measure—93.6%	Precision—94%, recall—87.66%, F score—88.11%	Precision—88.9%, recall—91.3%, F values—90.1%	Spams caught—94.47%, blocked ham, accuracy—98.88%	Threshold probability, accuracy—95%
Medium of Spam	Tweeter	Tweeter	Linguistic phrases	SMS	Blogs

Wang et al. [105] explored linguistic and behavioral features to detect the spams. This model's attention-based neural network model detects the review spam by distinguishingly using linguistic and behavioral features. In the feature extraction module, both behavioral features and linguistic feature vectors are calculated. The feature attention module calculates weighted feature vectors which are used to predict the spam probability. In [106], the authors presented an approach to detect the spam from short message service (SMS) using ML algorithms. For the detection and classification, a total of five models are used: (1) multinomial naïve Bayes; (2) SVM; (3) kth nearest neighbor; (4) random forest and; (5) AdaBoost with decision trees. As per the results, multinomial naïve Bayes spam detected 94.4% of spam which is the highest among all the five models. The block hams rate is 0.51% for Naïve Bayes and AdaBoost with decision trees. The accuracies are 98.88% and 98.86% of the multinomial Naïve Bayes and SVM respectively which are the highest compared to the other ML algorithms. The medium of spam through which it gets spread also performs a vital role in the detection with the use of ML algorithms. Therefore, we review the same for the existing works in Table 8.

3.5. Risk Assessment

Risk assessment provides a comprehensive view of the existing organization or system to obtain risk consequences, security risk [108] and countermeasures to deal with them. According to [109], risk assessment techniques are divided into two types of risk: (1) qualitative risk and; (2) quantitative risk. Qualitative risk is realized from policy direction and quantitative information, stakeholder knowledge and the history lessons for the system, risk profile and impact [110]. The conclusion derived from the qualitative assessment is more comprehensive and intellectual. Analytical hierarchy method [111], factor analysis method [112], a ranking method [110] and the delphi method [113] were proposed in the past for risk assessment. While quantitative risk is realized by the number of indicators such as probabilistic risk assessment [114], a number of induced equivalence profiles [115] also has a close connection with the system operability. Therefore, the result obtained by the quantitative risk assessment is more concise, clear and reliable. Correlation method [116], time series method [117] and cluster analysis [118] are the quantitative methods proposed by the researchers. Risk assessment using both the types of methods in the context of information security is subjective, vague and lacks the self-learning ability of the models which can be overcome by ML. In Table 9, we show the type of risk associated with the existing approaches and which algorithm is used to address it.

Eminagaoglu et al. [119] survey information security-related risk with the use of ML to prioritize the risk. In the first step, they collect qualitative evaluation regarding risk in the institution. In this survey, a total of six assets, ten threats and nine vulnerabilities are included. Qualitative scores from the respondents are collected from the questionnaires and it is analyzed by the ML classifiers. Overall risk can be identified as 'NO' if the scale rank of the respondent is from one to three and 'YES' if the scale rank of the respondent is from four to five. Dataset is made of 12 attributes and 1920 instances for learning the binary classifiers. In [120], the authors develop a tool, called RISKMON, which assesses the risk of a mobile application and uses an SVM learning algorithm to assess the risk rank.

Guntamukkala et al. [121] proposed an automated scheme based on integrity which helps the users to obtain online privacy policies. A contextual corpus of the online privacy policies was developed by the authors for training and testing purposes. With the use of text mining and ML techniques, this scheme evaluates the completeness of the online privacy policies and assesses the risk quantitatively. In [122], the authors presented an ML-based mechanism for managing information security with the use of the security metrics model. This model is used to get the arithmetic values for the security level. The results of the model are used in manufacturers' factories for assessing the security systems and for improving the controlling information risk. The authors in [123] proposed an ML-based approach to secure data in the cloud environment. In this approach, a Cloudsec module is used to reduce the risk of potential disclosure of the medical information. The SVM classifier is used to segment the image and data protection.

3.6. Malware Detection

In modern days, the most common method to launch attacks on modern computers, as well as network infrastructures are to use malware (worms, botnets, viruses, trojans). Antivirus software is the most common tools used to tackle different types of malware. To detect malware, three complementary, approaches are used: (1) signature-based detection; (2) anomaly-based detection and; (3) heuristics-based detection. Based on these three types of malware detection approaches we have analyzed in Table 10.

The signature-based detection approach scans and evaluates the type of its information from the file and, maps that information to jargons of virus signature that resides in its repository [124]. In this approach, the code is put under observation and approximate runtime behavior/pattern is predicted to detect the malware. In the static approach, many detection mechanisms are proposed such as annotated context-free graph [125], disassembled code [126], portable executable binary code [127] and honeypots [128]. Using this technique dynamic analysis of the malware is difficult to conceal.

The anomaly-based detection approach detects the malware by inspecting its runtime behavior. Techniques such as file print using n-gram [129], dynamic executable files [130], audit logs [131], function calls [132] and alert correlation framework [23] are used to detect malware based on its behavior. This approach can recognize only the presence of malware after malware code has been executed. Detection of the zero-day malware using this approach is difficult to achieve.

Heuristics-based detection uses ML and data mining techniques for malware detection. This approach overcomes the disadvantages of both the above malware detection approaches. This approach addresses the automatic monitoring of malware behavior and attempts to achieve the desired goal of detection. In the heuristics-based classification, two types of approaches can be identified: (1) clustering of behavior [133] and; (2) classification of behavior [134]. Das et al. [135] proposed an approach to capture malicious behaviors based on high-level semantics. The authors proposed a model based on benign patterns and system call patterns which generates the feature sets of known malware. Field programmable gate arrays allow the sharing of hardware for classifiers and runtime detection. Classifiers are used to identify the unknown samples of malware and benign software. In [136], the authors proposed a mechanism which identifies a malicious application on smartphones. This mechanism, in the static analysis phase, extracts different feature sets from the applications manifest and dalvik executable code, as mentioned in Table 10.

Gavrilut et al. [137] proposed an approach in which feature sets are enumerated for each binary file during training and testing datasets. In the ML framework, features are mapped into one side perceptrons and kernelized into one side perceptron. The F1 and F2 scores are used to combine the feature selection and training the ordinary size datasets containing the clean files and malware. In [138], the authors proposed an approach to identify malicious behavior based on the virtual memory access patterns in terms of a function call and system call. For system calls, the feature is selected using the F-score and topmost 10 percentage of F-score features for the training. Memory access in each function call is restricted to some memory, concluding in histograms with less than non-zero bins, which are used to train the classifiers.

Table 9. Review of risk assessment schemes.

Reference	[119]	[120]	[121]	[122]	[123]
Type of ML Algorithm	Random Forest, J48, K-Nearest Neighbor	SVM	Linear SVM, K-Nearest Neighbor, Random Forest	SVM	SVM, Fuzzy c-means clustering
Scheme Importance	Quantifiable risk assessed in a robust and reliable way	Automatically measure the risk induced by the user	Automated techniques to enumerate the integrity of the privacy policy and notifying the users about the important sections	Leads towards effective assistance and assessment to improve controlling information risk	Secure environment in cloud without revealing sensitive data
Limitation	Fuzzification, statistical, the numerical method can improve the performance	Doesn't address third-party applications, unauthorized access possible	Low degree of transparency may generate ambiguous results	Approach is designed for a single organization	Pixel texture feature can be added to enhance image segmentation
Performance Metrics	Accuracy—100%	Risk score	Accuracy—75%	Classifier Margin	Accuracy
Type of Risk	Qualitative Risk	Qualitative Risk	Quantitative Risk	Quantitative Risk	Qualitative Risk
Types of risk Identification	Security risk of the institution	Android mobile app risk	Privacy Policy Risk	Information risk	Disclosure of medical information

Table 10. Malware Detection Analysis (Notations: LR-Logistic Regression, SMO-Sequential Minimal Optimal, MLP- Multi-Layer Perceptron).

Reference	[135]	[136]	[137]	[138]	[139]
Type of ML Algorithm	J48, naïve bayes, SVM, LR, SMO, MLP	Linear SVM	Cascade one-sided perceptron, Cascade kernelized one-sided perceptron	Random Forest, Logistic Regression, SVM	SVM
Advantage	Able to get the high-level semantics of the malware	Able to protect user to install the application from an untrusted source	Non-stochastic version of algorithm enables parallelized training process to increase the speed	High detection accuracy against the kernel level rootkits and user level memory corruption report	Able to identify unknown families and behavior which are not present in the learning corpus

Table 10. *Cont.*

Reference	[135]	[136]	[137]	[138]	[139]
Limitation	Unable to detect kernel rootkits	When new code is loaded dynamic triggering is not possible	Accuracy is less when scaling up with the large datasets	For entire programme, Epochhistogram size should be chosen carefully which requires human effort	Relies on single program execution of malware binary
Performance Metrics	False positive rate, false negative rate, Accuracy- 99.7%	False detection, missing detection, accuracy- 93%	Sensitivity measure value, specificity measure value, accuracy measure value- 88.84%, True positives, false positives	False positives, true positives	Accuracy- 88%, confusion matrix
Type(s) of Malware detected	Backdoors, exploits, user-level rootkits, exploit, flooder, hack tools, net-Worm, Trojan, virus	Fake installer, DroidKungfu, Palnkton, opfake, GingerMaster, BaseBridge, Iconosys, Knim, FakeDoc, Geinimi, Adrd, DroidDream, LinuxLottor, GoldDream, MobileTx, FakeRun, Sendpay, Gappusin, Imlog, SMSreg	Backdoor, hack Tool, rootkit, Trojan, worms	Root kits	Worm, backdoors, trojans
Type of features employed to the classifiers for detection	Memory, network, file system, process- related system calls	Suspicious API calls, requests permissions, application components, filtered intents, network addresses, hardware features, used permission, restricted API calls	Binary type feature set	Architectural events, memory address, instruction mix	Frequency of contained string, string features (name & list of key-value pairs)

3.7. Testing Security Properties

Security properties play a key role in any distributed systems such as military infrastructure, banking, e-commerce, safety-critical autonomous systems, mobile ad-hoc and more [140]. Many analysis and modeling techniques have been proposed to make sure the correctness of the security protocol. These works aimed at validating the protocol specifications. A report [141] shows that errors and bugs during programming are common in the security-critical system which is to be identified and addressed properly. Therefore, automatic test drafting and proper execution methods are enticing to test real-time response and protocols in the security flaws. In addition, black-box testing is also another approach to verify specification and conformance of the protocol implementation, automation and formal modeling [142]. Existing testing techniques depend on human insights and, their skills and strategies, while traditional techniques do not deal beyond what is detailed in the specifications. With the use of ML, accurate testing of the properties would be possible. Vardhan et al. [143] proposed finite a state machine communicating over unbounded first-in–first-out (FIFO) channels with the use of an ML technique in order to verify the safety properties. The idea of the approach is to learn the set of reachable states rather than computing them by assigning transition relations.

ML techniques learn the reachable first states and then verify the safety properties by checking the unsafe states in the reachable states. In the case that the arrangement of states learned closed under the change connection infers. The scholarly arrangement of states contains every single reachable set which does not contain any unsafe state, at that point, it is inferred that framework fulfills the security properties. For learning positive and negative samples, RPNI algorithm [144] is used. Shu et al. [145] proposed a new ML technique to automatically test protocol implementation security characteristics. The security protocol model is a symbolic parameterized extended finite-state model and, message confidentiality is a property to be investigated with the help of this model. If the attacker can obtain a confidential message with the use of prior knowledge, there is a security violation of the proposed model. The goal of the message confidentiality testing is to discover the security contravention with the help of black-box testing. At the time of testing a component of the model is kept under test and so is the anticipated behavior of the component. The behavior is represented in terms of finite state machine (FSM) trace which calculates the estimation, and updates based on the learning algorithm that covers the target implementation. After forming the new estimation, the validation algorithm is executed which calculates the reachability of the graph and searches for the security violation. If the violation is confirmed in the security properties that means false positive introduced and experiment trace on the black box and if the trace is confirmed then it will be a FAIL claim. If no violation is found and evaluation is similar to the black box, the process terminates with the PASS result. In [146], the authors presented a formal fuzz testing and ML-based communication protocol for security flaws detection. Authors adopt the FSM protocol model and examine two conventional techniques for the protocol synthesis: (1) active black box-checking algorithm; (2) passive trace minimization algorithm. As discussed earlier, the behavior of FSM is updated as more traces are covered by the supervised ML algorithm. To take control of input–output (I/O), a proxy has been developed which connects the client to the server. In the login phase, the model has been synthesized with the protocol involving approximately 50 states and 70 transitions. The aim of the fuzzy function is to search the series of inputs that crash the client process.

4. Review of Adversarial Attacks on Machine Learning

4.1. Machine Learning Vulnerability Analysis and Threat Model

We quote the definition of software vulnerability described by Mohammad et al. in [147]:

> "Software vulnerability is an instance of a flaw, caused by a mistake in the design, development or configuration of software such that it can be exploited to violate some explicit or implicit security policy."

Vulnerability is the root cause of security and privacy breach of any system. In this analysis, we describe different attacks in terms of security and privacy perspective.ML techniques are broadly applied in security and privacy-oriented operations such as malware detection, pattern recognition, spam detection, pattern recognition, homomorphic encryption [148,149] and privacy preservation and statistical analysis of a database [150,151]. ML has become a promising approach in order to provide automation in security or privacy breach detection. As above mentioned, from the perspective of security and privacy, it is impossible to make a system that addresses these two aspects. ML techniques also have weaknesses and vulnerabilities. Therefore, we have demonstrated an attack surface as well as the possible scope of the defense in the ML life cycle in Figure 3. It is designed to illustrate the characteristics of the possible attacks under adversarial settings. While the middle layer in Figure 3 represents data pre-processing, feature extraction and model training phase of the ML classifier. The top layer demonstrates the countermeasures over the adversarial setting for the ML classifier.

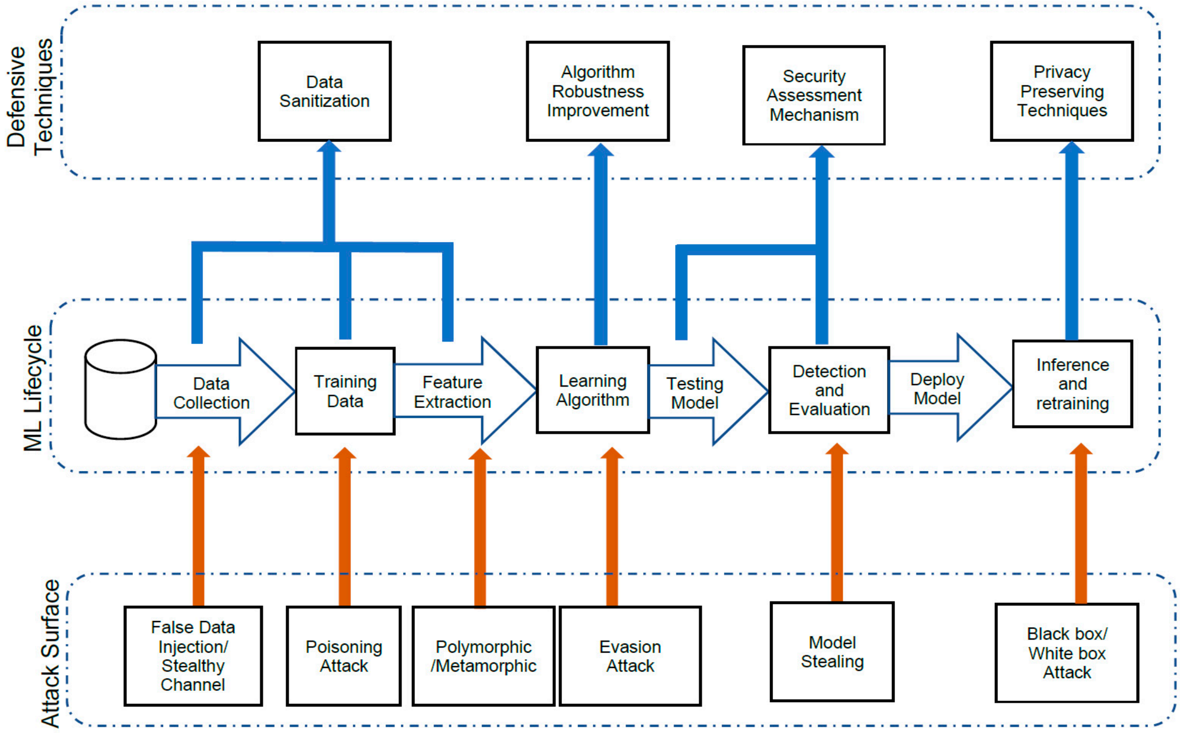

Figure 3. Threat model of the machine learning process.

As demonstrated in Figure 3 in the bottom layer, the attacker can access the ML classifier by false data injection and stealthy channel attacks. The training phase is vital for ML classifiers to realize a specific classification with respect to a dataset. The poisoning attack has rattled the integrity and availability of the ML models by injecting the adversarial samples into the training datasets. In a real-world scenario, ML-based system training data is highly protected with confidentiality therefore, homomorphic scheme is one that can transform the one feature vector entity into another feature vector entity. The evasion attacks are proposed to imperil the ML model security by modifying key features of the ML algorithm and gain the authority of the model. After deployment of the ML classifier attacker may exploit a stolen model to detect negotiable adversarial models that can deceive classification by the authentic model. For this activity, attackers mount prediction application program interfaces by sending repeated queries. In the inference phase, according to the degree of understanding knowledge in the attacker model, it is classified into two groups, particularly white-box, and black-box attack. Sophisticated and strong attackers can launch a white-box attack by downloading and accessing the ML models and other data, while black-box attacks can launch by weak attackers by using APIs and filling inputs.

In the top layer of Figure 3, the possible defensive techniques for the ML lifecycle over the adversarial techniques to defend the ML classifiers in various stages are demonstrated. Data sanitization is one of the approaches to protect the purity of the training data by isolating the adversarial sample from the original sample and reject the adversarial sample. Another adequate technique to improve the robustness and security of the algorithms can be used which evenly distributed feature weights of classifiers. Also to improve the robustness of the algorithm retraining the ML classifiers with the adversarial samples so that newly trained classifiers are able to detect anomalies in the testing phase. While security assessment scheme involves a risk assessment scheme involves in Table 9 in order to assess the security of ML classifiers to protect the possible threats against the attacker. With security assessment in the big data era, confidentiality and the privacy of the data is also an essential and vital concern in the defense techniques. Therefore to provide data privacy with the existence of the sophisticated attacks differential privacy and homomorphic encryption techniques are used.

4.1.1. Categorizing of Attack Properties

In this section, we demonstrate different properties of the attack and we organized accordingly in ML perspective in Figure 4.

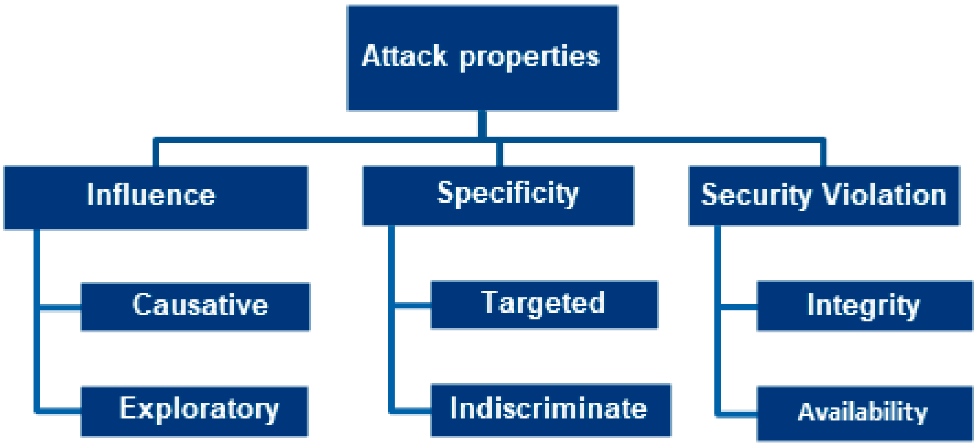

Figure 4. Taxonomy of attack properties on machine learning.

1. Influence

Influence property describes the attacker's potential to influence the machine level model at each level. Causative attacks to leverage the training data by taking control of the datasets exploratory attacks do not affect the training data but influence the classifier by aiming misclassification in offline mode or using probing techniques.

2. Specificity

Specificity property indicates the objective of the attacker's intention over the ML classifiers. Targeted attacks aim at the classifiers that degrade classifiers' performance when particular conditions are fulfilled. Indiscriminate attacks cause damage to the classifiers in an assorted manner with a large number of instances.

3. Security Violation

Security property shows the level of security violations done by the attackers. Integrity attacks allow a malicious instance as a genial instance by poisoning the filters. This gives incorrect results in the form of a false negative rate. Availability attacks are caused by DoS attacks in which genial instances are identified as malicious by poisoning the filters and as a result, it will provide incorrect false-positive rates.

To get an idea about the security in ML, one should address primary issues in adversarial settings with a proactive approach [152]:

- Diagnose probable vulnerabilities during training and classification in the machine learning algorithm.
- Model the types of attacks that coincide in order to recognize different threats and to evaluate the impact on the victim.

The results observed in [152] urge corrective measures to enhance the security of ML algorithms and training data against adversarial attacks.

A proactive approach to ML algorithms needs reverse engineering as shown in Figure 5. Such an approach does not represent obscure or evolving parts of the adversary. Without a doubt, it can prompt an enhanced level of security by deferring each progression of the receptive weapons contest. Since its constraints ought to sensitize the foe to apply more noteworthy exertion to discover new vulnerabilities.

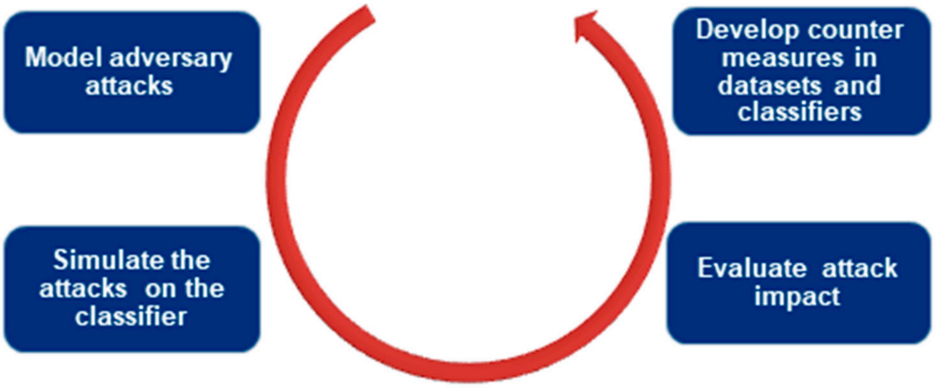
Model adversary attacks
Develop counter measures in datasets and classifiers
Simulate the attacks on the classifier
Evaluate attack impact

Figure 5. Proactive approach to machine learning attacks.

In Figure 6, we describe the level of the scenario for the classifier in the context of the training dataset, feature sets, and algorithm of the classifiers. If we think with the adversarial point of view, knowledge is required based on these three evasion scenarios. Therefore, based on these scenarios, the other four scenarios can be concluded as shown in Figure 6. We align each scenario with Table 11.

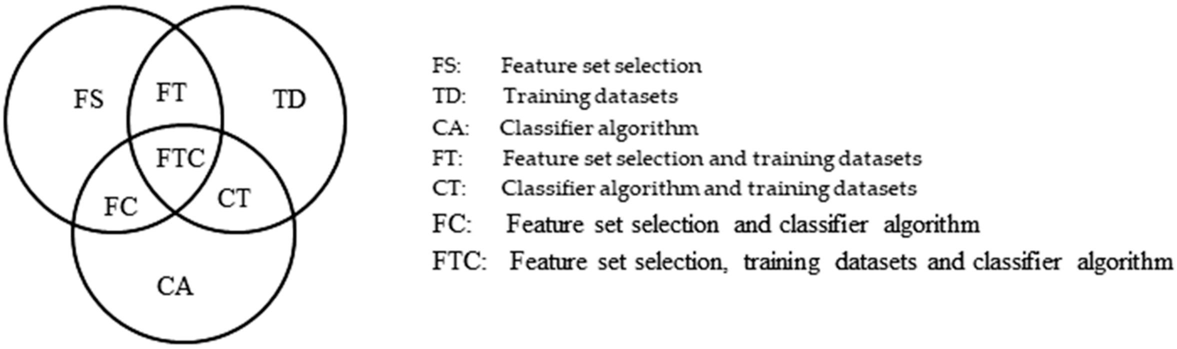
FS
FT
TD
FTC
FC
CT
CA
FS: Feature set selection
TD: Training datasets
CA: Classifier algorithm
FT: Feature set selection and training datasets
CT: Classifier algorithm and training datasets
FC: Feature set selection and classifier algorithm
FTC: Feature set selection, training datasets and classifier algorithm

Figure 6. Levels of evasion scenarios for the classifier system [153].

Table 11. Influence of attacker classifier by attacker type.

		Attacker Type		
		Weak Attacker	**Strong Attacker**	**Sophisticated Attacker**
Classifier Level	FS	•	•	•
	TD	•	•	•
	CA			•
	FT		•	•
	CT			•
	FC			•
	FTC			•

4.1.2. Attackers' Category

For any kind of attacker on any kind of system, there is a requirement of comprehensive knowledge, phase of attack contamination and budgets for modeling the proper attacks. From ML point of view, as described in [154], attackers' familiarity with the training data, feature sets, decision function, learning algorithm, and parameters varies from system to system. In order to benchmark the ML model during the designing and development phase, we have considered a white box testing scenario for which we have defined the role of sophisticated attacker to map with the evasion scenario in the model.

1. Weak attacker

A weak attacker does not have the knowledge of the statistical properties of training or labels. This kind of attacker tries to poison the training datasets by adding fake labels. Thus, a weak attacker can poison the properties feature sets and training sets in the classifier system.

2. Strong attacker

This type of attacker can access the dataset and influence features of the datasets. This category of the attacker cannot influence directly on classifiers, rather it uses publicly available malware poisoning. A strong attacker can poison properties such as feature set, training data, and feature set and training set in the classifier.

3. Sophisticated attacker

A sophisticated attacker has knowledge of the algorithm and parameters to run that algorithm. This type of attacker has sufficient economic resources and, can manipulate all the training datasets and feature data with the use of malware. This attacker can influence all types of evasion scenarios, as shown in Table 11.

4.1.3. Attacks on Machine Learning by its Security Property

This section is described based on the taxonomy of the attack properties bifurcate different types of attacks possible on the machine learning classifiers.

1. Causative Attacks

In causative attacks, an adversary impacts the training data which drives towards misclassification. On training, a data attacker has a different type of impact from capricious dominations to prejudice dominations, over some portion of information generation [155].

A. Causative Integrity Attack

In this type of attack, an adversary makes spam slip past the classifier as a false negative by employing control overtraining. An example of a causative integrity attack is the label flipping attack. The goal of a causative integrity attack is to include false labels into training data by flipping the labels. Attackers are able to modify legitimate labels and introduce them as malicious and vice-versa in this attack. To do this, common methods are used to collect malware data with the use of honeypots and botnets.

Attack scenario: in [156], the authors proposed a model of adverse flipping attack in which they assumed that attackers are able to manipulate labels maliciously to mislead the classifier over the non-malicious datasets. Thus, it preserves its generalization on malicious datasets. By doing so, the hyperplane of the classifiers will be migrated and as a result, the attacker deviates generalization of the classifiers from non-malicious data sets. For modeling attacks, the authors used real-world datasets and synthetic data sets. For synthetic datasets, 200 training samples were randomly selected and, test errors are performed on flip L = 20 labels and disjointed set of 800 samples from the training data. On synthetic datasets, for decision boundary, linear and radial basis function (RBF) kernel is used with a value of C = 1 (soft margin of influence control for support vector) and $\gamma = 0.5$ (similarity measure). When the RBF kernel is applied, the efficiency of SVM is influenced by carefully choosing labels from the training datasets. Therefore, it is depicted that a change in the SVM algorithm has a notable impact on the results. Thus, to get the maximum impact in the results from the attackers' perspective, it requires the knowledge of training datasets.

B. Causative Availability Attack

The main goal of the causative availability attack is to use token-based features to train the classifiers maliciously. In the attackers' perspective, they can add malicious features into the training instance which causes filter blockage of non-malicious features in this type of attack.

- Attack scenario: the authors in [157] discussed attacks against a spam Byers in terms of indiscriminate and targeted dictionary attacks. In indiscriminate dictionary attacks, the email contains words, which are liable to show authentic messages. Accordingly, these types of email words are incorporated into information preparation and as a result, the classifiers will classify authentic emails as spam. While in the targeted dictionary attacks, the adversary considers the knowledge of a particular email instead of reading it from the recipient. Hence, the impact is limited because it is word specific. In [158], the authors presented allergy attacks on the autograph worm generation system. This attack is divided into two phases. In the beginning, based on the behavioral patterns during scanning, it distinguishes tainted nodes from the network. In the second phase, it analyses the traffic from tainted nodes and, deduces the blocking rules from the observed behavioral patterns. Thus, the autograph is persuaded by the tainted node, which is contaminated by scanning the network. The tainted node sends forged packets which results in the DoS and blocking non-malicious access from the autograph.

2. Exploratory Attack

During the exploratory attack, an adversary modifies the spam structure with the use of polymorphic, metamorphic or rootkits that use different types of obfuscation techniques. These types of scenario attacks avoid the direct influence of the classifier on itself. These types of attacks are either targeted or indiscriminate.

A. Exploratory Integrity Attack

The goal of an exploratory integrity attack is to mask intrusion by resembling the statistical properties of network traffic in the training data is calculated by the classifiers. Adversaries have a direct influence on the ML classifiers as the sophisticated types of attackers have a perfect knowledge of classifiers while the other attackers have knowledge of feature representation and types of classifiers, but it does not know whether classifiers have learned; thus, it will not be able to calculate the discrimination function.

- Attack scenario: in [159], the authors discussed a model of mimicry attack with the use of a gradient descent method over the neural network and SVM classifier. The authors modeled attacks from both sophisticated and strong attackers' point of view. Feature values will be changed by the attacks prior to the attack point flip labels. This attack is applied to handwritten grayscale images using the SVM classifier and to portable document format (PDF) document using neural network and SVM classifier. The authors used the feature for malicious PDF, which was extracted in [160].

In handwritten images, authors have considered perfect knowledge scenarios from the attacker's point of view. The grayscale pixel values were changed to modify the handwritten digits. In this attack, the targeted classifier was an SVM consisting of linear kernel functionality. The authors chose 100 random training samples for applying the attack. In the given gradient attack, digit '3' is misclassified as digit '7'. Without using a mimic component $\lambda = 0$, gradient descent quickly gets decreased, but not able to classify digit '3' as digit '7' after 500 iterations. While using a mimic component $\lambda = 10$, gradient the attacked image precisely resemble due to mimicry term more favorable to the target class. So when mimic is used the discriminant function g(x) tends to decrease more gracefully.

For the PDF sample, the authors have modeled attacks with attackers having perfect or limited knowledge. For a limited knowledge case, the false-negative rate computed corresponds to false postives = 0.5%. When the false-negative rate increases with dmax (maximum distance from the original attack sample dmax ϵ [0, 50]), PDF is progressively modified. When dmax = 0, there is no change in false-negative rate as PDF is unmodified. For linear SVM, without the use of mimicry component ($\lambda = 0$) in perfect knowledge and limited knowledge cases to 1 to 0.75 respectively with 5 to 10 modification. For RBF kernel with mimicry component ($\lambda = 0$), false-negative rates for PK and LK cases are 0.8 and 0.6 respectively with 15 modifications. While comparing to both, SVM and neural networks, the neural network is more robust against the proposed attack. Furthermore, in the absence of mimicry component ($\lambda = 0$), the false-negative rate is 0.2 with 50 modifications for a neural network in PK scenario. While in the presence of mimicry component ($\lambda = 500$) for a linear SVM, false-negative rate increases slowly as compared to RBF kernel and neural network, for both PK and LK scenarios. The neural network is more vulnerable in the presence of mimicry component ($\lambda = 500$) as false-negative rate 1 requires only 20 modifications in PK scenario and false-negative rate 0.5 requires 50 modifications in LK scenario.

B. Exploratory Availability Attack

The main goal of an exploratory availability attack is to set of points that are misclassified by the learner to launch a DoS attack. Attackers require knowledge of production learners during the attack.

- Attack scenario: in [147], the authors proposed a mechanism for threatening statistical traffic analysis by emulating the class of traffic which mimics another class. This method focuses on the detection system. The packets are modified in real-time which reduces the accuracy of classifiers. Classifiers achieve the accuracy of 98.4% on unmodified data, while it is reduced to just 4.5% after the attack.

4.2. Practical Feasibility of Attacks

Attacks describe in Section 4.1.3, highlights several practical feasibility and consideration that must be addressed to craft effective attack against a machine learning system. As mentioned above that weak attackers do not have knowledge about the statistical properties of the training data or model. While strong attackers don't have knowledge about classifiers. This type of attacker can practically craft black box types of attacks. The main strategy followed by the weak attacker in these types of attacks is a substitute synthetic dataset to drive the classifier for misclassification [161]. Weak attacker embeds a portion of benign code into a malicious app by misusing Manifest.xml file configuration for any malware detection application. By such type of practice as mentioned in [162], AndroidManifest.xml file of Polaris Office misclassified as a benign file. Strong attacker mainly influences the feature sets such as the address of the designated system is challenging to spoof. Xiao et al. [163], represent the attack in which by injecting a maximum number of poisoning point into training data sets and maximize the classification error of the algorithm. While sophisticated attackers have knowledge of algorithm and parameters to run the algorithm. Such type of practice is regularly used in the lab for the penetration testing of the machine learning classifiers. As mentioned in [164], using dynamic code loading modify the runtime behavior of the applications and mislead the classifiers.

4.3. Adversarial Defense Techniques

In the defense for the ML classifiers, designers proposed two types of mechanisms that is proactive defense and reactive defense [152]. In the proactive designer select any one of the adversarial models. Then adversary launches the penetration testing on the model and analyses the impact of the attacks. After the designer proposed countermeasures towards adversary while designing a classifier. While in the reactive defense designer analyses new added samples and corresponding attack results after the attack over the ML classifier. Then the designer proposed defending mechanisms for the ML classifiers. To take countermeasures in the training phase designer try to ensure data purity and improve the robustness of the algorithm [165]. While countermeasures in the inference phase only focus on the robustness of the classifier. In [166] proposed an ensemble method, which improves the robustness of the classifier by fulfilling the availability/integrity type of security property. To provide security and privacy of the data cryptographic technology and differential privacy is used [167,168]. These schemes ensure blocking data leakage and reduce sensitive outputs of the classifiers, which are the impact on data security and privacy. Therefore, by using these types of adversarial defense techniques designer is defending the ML classifiers. In Table 12, we represent the review of both proactive and reactive types of defense techniques.

Table 12. Adversarial defense techniques analysis over security properties (reject on negative impact (RONI)).

Reference	[165]	[166]	[167]	[168]	[169]	[170]
Technique	Data Sanitization	Ensemble method	Differential privacy	Holomorphic encryption	Adversarial training	Defense distillation
Type of security property	Protect availability and integrity	Protect availability and integrity	Preserve privacy and integrity	Preserve privacy and integrity	Protect availability and integrity	Protect availability and integrity
Advantage	Based on RONI sanitization rejects samples which include a negative impact	Ensemble the training data with perturbations transferred from other models	Give protection against strong attacker over training mechanism and model parameters	Able to protect data privacy multi-party computational environment and directly process encrypted data	Provides the examples of adversarial training and labels during training to identify adversarial in future	Reduces sensitivity of networks to adversarial manipulation of their inputs also leverages the resilience to adversarial crafting
Disadvantage	RONI defense fails to detect a focused attack	Not considered black box adversaries that attack a model via other means	Differential privacy accuracyis less	This approach is designed only for the horizontal data objects only	Ensemble approach provides limited resistance to adversarial perturbation	Add only restricted amount of features also gradually cannot change the features
Type of approach	Reactive	Reactive	Proactive	Proactive	Reactive	Proactive

5. Conclusions

New threats caused by cyber-attacks can damage critical data infrastructure because of machine learning in the security applications highly dependent on the data quality. Using machine learning-based methods in security applications faces a challenge the performance of recognizing an adversarial sample by collecting and predicting adversarial samples. Hence we conclude that the new models are becoming a research point from attacker and designer perspective. With the rapid increase in security events security in machine learning-based decision systems in adversarial environments opens a door for the new research area. In some cases, malicious users can simply increase false-negative rates and minimizing false-positive rates by a proportional amount, cleverly make sure that the overall error rate remains the same and attack is unnoticed which can give attackers some leverage in sophisticated attacks. This kind of issue there needs to be explored to detect attacks efficiently on ML-based systems. Regardless of the data privacy field, great advancement in existing methods of data privacy suffer from modest performance due to complex operations on a huge number of parameters of machine learning algorithms. Therefore extremely efficient privacy-preserving methods need to be investigated in the adversarial environmental setting. Observation made related performance tradeoff between accuracy and scalability for the machine learning classifiers. For example for any security application designing informal decision made on which approach to use when. But even though having more weak labels does not imply that classifiers' accuracy will eventually reach a precise accuracy. Therefore, it is worth to infuse humans or utilizing transfer learning to make additional changes. This type of decision is made by an experiment, but an important question is whether, overall, there is a need to design and craft secure machine learning algorithms that way which can balance three aspects that are performance overhead, security optimization, and performance generalization.

Few conventional techniques rely on known threats due to the precedence of vulnerabilities. Therefore, with the evolvement of the threats, there is a need to upgrade the detection techniques to counter the new generation threats. In this survey, we present the taxonomy of threats that infer the overall characteristics, structure, types and spreading mechanism of different types of malware. We discuss different types of security applications where machine learning is used to leverage the fulfillment of current world requirements from security and privacy perspective. We discuss and compare different types of machine learning models from the security and privacy point of view. Moreover, we highlight possible limitations of the proposed approaches and challenges involved with the same. We include scenarios of evasion for a machine learning system from the adversarial point of view. In addition, we align the attack scenarios on the machine learning classifiers with the attackers' knowledge. Furthermore, we illustrate the attacks aimed at the machine learning classifiers and algorithms that can cause damage in the context of the security properties of the model. Moreover, we review defense techniques for the machine learning classifiers which preserves the machine learning security properties. We find that sophisticated attacks can exploit the machine learning-based malware detectors with extreme severity. Therefore, it is imperative to protect machine learning-based security solutions and address their vulnerabilities. The sole purpose of this survey is to direct the security researchers in devising more secure, reliable and effective models.

Author Contributions: This survey articles was the result of contributions from all authors more or less equally. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lee, Y.S.; Choi, S.S.; Choi, J.; Song, J.S. A Lightweight Malware Classification Method Based on Detection Results of Anti-Virus Software. In Proceedings of the 2017 12th Asia Joint Conference on Information Security (AsiaJCIS), Seoul, Korea, 10–11 August 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 5–9.

2. ESMA. 2016 Annual Report (IEEE). Available online: https://www.ieee.org/about/annual-report.html (accessed on 4 October 2019).
3. Zhang, X.; Tan, Y.A.; Liang, C.; Li, Y.; Li, J. A Covert Channel over VoLTE via Adjusting Silence Periods. *IEEE Access* **2018**, *6*, 9292–9302. [CrossRef]
4. Meng, W.; Tischhauser, E.W.; Wang, Q.; Wang, Y.; Han, J. When Intrusion Detection Meets Blockchain Technology: A Review. *IEEE Access* **2018**, *6*, 10179–10188. [CrossRef]
5. Li, J.; Sun, L.; Yan, Q.; Li, Z.; Srisa-an, W.; Ye, H. Significant permission identification for machine learning based android malware detection. *IEEE Trans. Ind. Inform.* **2017**, *14*, 3216–3225. [CrossRef]
6. Shen, J.; Wang, C.; Li, T.; Chen, X.; Huang, X.; Zhan, Z.H. Secure Data Uploading Scheme for a Smart Home System. *Inf. Sci.* **2018**, *453*, 186–197. [CrossRef]
7. Sun, Z.; Zhang, Q.; Li, Y.; Tan, Y.A. DPPDL: A Dynamic Partial-Parallel Data Layout for Green Video Surveillance Storage. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 193–205. [CrossRef]
8. Buczak, A.L.; Guven, E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1153–1176. [CrossRef]
9. M.Labs, Malwarebytes. State of Malware Report. p. 11. Available online: https://www.malwarebytes.com/pdf/white-papers/stateofmalware.pdf (accessed on 4 October 2019).
10. Balzarotti, D.; Cova, M.; Felmetsger, V.; Jovanovic, N.; Kirda, E.; Kruegel, C.; Vigna, G. Saner: Composing static and dynamic analysis to validate sanitization in web applications. In Proceedings of the 2008 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 18–22 May 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 387–401.
11. Shar, L.K.; Briand, L.C.; Tan, H.B.K. Web Application Vulnerability Prediction Using Hybrid Program Analysis and Machine Learning. *IEEE Trans. Dependable Secur. Comput.* **2015**, *12*, 688–707. [CrossRef]
12. Rohlf, C.; Ivnitskiy, Y. The security challenges of client-side just-in-time engines. *IEEE Secur. Priv.* **2012**, *10*, 84–86. [CrossRef]
13. Shen, J.; Zou, D.; Jin, H.; Yang, K.; Yuan, B.; Li, W. Security OF Cyberspace A Protective Mechanism for the Access Control System in the Virtual Domain. *China Commun.* **2016**, *13*, 129–142. [CrossRef]
14. Navarro-Arribas, G.; Torra, V. Information fusion in data privacy: A survey. *Inf. Fusion* **2012**, *13*, 235–244. [CrossRef]
15. Yuan, J.; Member, S.; Yu, S. Privacy Preserving Back-Propagation Neural Network Learning Made Practical with Cloud Computing. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 212–221. [CrossRef]
16. Smith, J.E.; Clark, A.R.; Staggemeier, A.T.; Serpell, M.C. A genetic approach to statistical disclosure control. *IEEE Trans. Evol. Comput.* **2012**, *16*, 431–441. [CrossRef]
17. Matthews, G.J.; Harel, O. Data confidentiality: A review of methods for statistical disclosure limitation and methods for assessing privacy. *Stat. Surv.* **2011**, *5*, 1–29. [CrossRef]
18. Niksefat, S.; Kaghazgaran, P.; Sadeghiyan, B. Privacy issues in intrusion detection systems: A taxonomy, survey and future directions. *Comput. Sci. Rev.* **2017**, *25*, 69–78. [CrossRef]
19. Axelsson, S. The Base-Rate Fallacy and the Difficulty of Intrusion Detection. *ACM Trans. Inf. Syst. Secur.* **2000**, *3*, 186–205. [CrossRef]
20. Shon, T.; Moon, J. A hybrid machine learning approach to network anomaly detection. *Inf. Sci. (NY)* **2007**, *177*, 3799–3821. [CrossRef]
21. Bhoiwala, J.P.; Jhaveri, R.H. Cooperation Based Defense Mechanism against Selfish Nodes in DTNs. *ACM Int. Conf. Proc. Ser.* **2017**. [CrossRef]
22. Jhaveri, R.H. *Secure Routing in Mobile Ad-Hoc Networks: Attacks and Solutions*; LAMBERT Academic Publishing: Saarbrücken, Germany, 2017.
23. Vidal, J.M.; Orozco, A.L.S.; Villalba, L.J.G. Alert correlation framework for malware detection by anomaly-based packet payload analysis. *J. Netw. Comput. Appl.* **2017**, *97*, 11–22. [CrossRef]
24. Cohen, Y.; Hendler, D.; Rubin, A. Detection of malicious webmail attachments based on propagation patterns. *Knowl. Based Syst.* **2018**, *141*, 67–79. [CrossRef]
25. Paxson, V. Bro: A system for detecting network intruders in real-time. *Comput. Netw.* **1999**, *31*, 2435–2463. [CrossRef]
26. Jhaveri, R.H.; Patel, N.M. A sequence number based bait detection scheme to thwart grayhole attack in mobile ad hoc networks. *Wirel. Netw.* **2015**, *21*, 2781–2798. [CrossRef]

27. Jhaveri, R.H.; Patel, N.M. Attack-pattern discovery based enhanced trust model for secure routing in mobile ad-hoc networks. *Int. J. Commun. Syst.* **2017**, *30*, e3148. [CrossRef]
28. Jhaveri, R.H. MR-AODV: A solution to mitigate blackhole and grayhole attacks in AODV based MANETs. In Proceedings of the 2013 Third International Conference on Advanced Computing and Communication Technologies (ACCT), Rohtak, India, 6–7 April 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 254–260.
29. Jhaveri, R.H. *Reliable Approach to Prevent Blackhole and Grayholes Attacks in Mobile ad-hoc Networks*; IET Editorial Book; IET: London, UK, 2013; pp. 261–280.
30. Jhaveri, R.H.; Patel, N.M.; Jinwala, D.C.; Ortiz, J.H.; de la Cruz, A.P. *A Composite Trust Model for Secure Routing in Mobile Ad-Hoc Networks*; Ortiz, J.H., de la Cruz, A.P., Eds.; IntechOpen: London, UK, 2017; pp. 19–45.
31. Overill, R.E. How Re (Pro) active Should an IDS be? In Proceedings of the 1st International Workshop on Recent Advances in Intrusion Detection (RAID), Louvain-la-Neuve, Belgium, 14–16 September 1998; pp. 1–6.
32. Francisco, S.; Martin, D.; Schulman, A. Infranet: Circumventing web censorship and surveillance. In Proceedings of the 11 the USENIX Security Symposium, Berkeley, CA, USA, 5–9 August 2002.
33. Rhodes, B.C.; Mahaffey, J.A.; Cannady, J.D. Multiple self-organizing maps for intrusion detection. In Proceedings of the 23rd National Information Systems Security Conference, Baltimore, MD, USA, 16–19 October 2000; pp. 16–19.
34. Yamada, A.; Miyake, Y. Intrusion detection system to detect variant attacks using learning algorithms with automatic generation of training data. In Proceedings of the IEEE Coding and Computing, Las Vegas, NV, USA, 4–6 April 2005; IEEE: Piscataway, NJ, USA, 2005; pp. 1–6.
35. Lippmann, R.P.; Cunningham, R.K. Improving intrusion detection performance using keyword selection and neural networks. *Comput. Netw.* **2000**, *34*, 597–603. [CrossRef]
36. Wong, W.T.; Hsu, S.H. Application of SVM and ANN for image retrieval. *Eur. J. Oper. Res.* **2006**, *173*, 938–950. [CrossRef]
37. Borges, P.; Sousa, B.; Ferreira, L.; Saghezchi, F.B.; Mantas, G.; Ribeiro, J.; Simoes, P. Towards a Hybrid Intrusion Detection System for Android-based PPDR terminals. In Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal, 8–12 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1034–1039.
38. Karthick, R.R.; Hattiwale, V.P.; Ravindran, B. Adaptive network intrusion detection system using a hybrid approach. In Proceedings of the IEEE Fourth International Conference on Communication Systems and Networks, Bangalore, India, 3–7 January 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 1–7.
39. Freund, Y.; Schapire, R.E. Experiments with a new boosting algorithm. In Proceedings of the ICML'96 Proceedings of the Thirteenth International Conference on International Conference on Machine Learning, Bari, Italy, 3–6 July 1996; pp. 148–156.
40. Bauer, E.; Kohavi, R.; Chan, P.; Stolfo, S.; Wolpert, D. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Mach. Learn.* **1999**, *36*, 105–139. [CrossRef]
41. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
42. Parmanto, B.; Munro, P.W.; Doyle, H.R. Improving committee diagnosis with resampling techniques. *Adv. Neural Inf. Process. Syst.* **1996**, *8*, 882–888.
43. Ma, T.; Wang, F.; Cheng, J.; Yu, Y.; Chen, X. A Hybrid Spectral Clustering and Deep Neural Network Ensemble Algorithm for Intrusion Detection. *Sensors (Basel)* **2016**, *16*, 1701. [CrossRef]
44. Fries, T. A fuzzy-genetic approach to network intrusion detection. In Proceedings of the GECCO'08 10th Annual Conference Companion on Genetic and Evolutionary Computation, Atlanta, GA, USA, 12–16 July 2008; pp. 2141–2146.
45. Tsai, C.F.; Lin, C.Y. A triangle area based nearest neighbors approach to intrusion detection. *Pattern Recognit.* **2010**, *43*, 222–229. [CrossRef]
46. Van, N.T.; Thinh, T.N.; Sach, L.T. An anomaly-based network intrusion detection system using Deep learning. In Proceedings of the 2017 International Conference on System Science and Engineering (ICSSE), Ho Chi Minh City, Vietnam, 21–23 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 210–214.
47. Viegas, E.K.; Santin, A.O.; Oliveira, L.S. Toward a reliable anomaly-based intrusion detection in real-world environments. *Comput. Netw.* **2017**, *127*, 200–216. [CrossRef]

48. Ghanem, K.; Aparicio-Navarro, F.J.; Kyriakopoulos, K.G.; Lambotharan, S.; Chambers, J.A. Support Vector Machine for Network Intrusion and Cyber-Attack Detection. In Proceedings of the 2017 Sensor Signal Processing for Defence Conference (SSPD), London, UK, 6–7 December 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–5.
49. Al-Yaseen, W.L.; Othman, Z.A.; Nazri, M.Z.A. Hybrid Modified K -Means with C4.5 for Intrusion Detection Systems in Multiagent Systems. *Sci. World J.* **2015**. [CrossRef]
50. Ross, J.; Morgan, Q.; Publishers, K. Book Review: C4. 5: Programs for Machine Learning. *Mach. Learn.* **1994**, *16*, 235.
51. Abadeh, M.S.; Habibi, J.; Barzegar, Z.; Sergi, M. A parallel genetic local search algorithm for intrusion detection in computer networks. *Elsevier Eng. Appl. Artif. Intell.* **2007**, *20*, 1058–1069. [CrossRef]
52. Giacinto, G.; Perdisci, R.; del Rio, M.; Roli, F. Intrusion detection in computer networks by a modular ensemble of one-class classifiers. *Inf. Fusion* **2008**, *9*, 69–82. [CrossRef]
53. Gajin, V.T.S. Ensemble classifiers for supervised anomaly-based network intrusion detection. In Proceedings of the 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 7–9 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 13–19.
54. Hansen, J.V.; Lowry, P.B.; Meservy, R.D.; McDonald, D.M. Genetic programming for prevention of cyberterrorism through dynamic and evolving intrusion detection. *Decis. Support Syst.* **2007**, *43*, 1362–1374. [CrossRef]
55. Jhaveri, R.H. DoS Attacks in Mobile Ad Hoc Networks: A Survey DoS Attacks in Mobile Ad-hoc Networks: A Survey. In Proceedings of the 2012 Second International Conference on Advanced Computing & Communication Technologies, Rohtak, Haryana, India, 7–8 January 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 535–541.
56. Prichard, J.J.; Macdonald, L.E. Cyber Terrorism: A Study of the Extent of Coverage in Computer Security Textbooks. *J. Inf. Technol. Educ.* **2004**, *3*, 279–289.
57. Gunawardena, S.H.; Kulkarni, D.; Gnanasekaraiyer, B. A Steganography-based framework to prevent active attacks during user authentication. In Proceedings of the 2013 8th International Conference on Computer Science & Education, Colombo, Sri Lanka, 26-28 April 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 383–388.
58. Allen, J.; Gomez, L.; Green, M.; Ricciardi, P.; Sanabria, C.; Kim, S. Social Network Security Issues: Social Engineering and Phishing Attacks. In Proceedings of the Student-Faculty Research Day, CSIS, White Plains, NY, USA, 4 May 2012; pp. 1–7.
59. Applegate, S.D. Social engineering: Hacking the wetware! *Inf. Secur. J.* **2009**, *18*, 40–46. [CrossRef]
60. Khonji, M.; Iraqi, Y.; Jones, A. Phishing detection: A literature survey. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 2091–2121. [CrossRef]
61. Li, Y.; Xiao, R.; Feng, J.; Zhao, L. A semi-supervised learning approach for detection of phishing webpages. *Optik* **2013**, *124*, 6027–6033. [CrossRef]
62. Smadi, S.; Aslam, N.; Zhang, L. Detection of online phishing email using dynamic evolving neural network based on reinforcement learning. *Decis. Support Syst.* **2018**, *107*, 88–102. [CrossRef]
63. Hamid, I.R.A.; Abawajy, J.H. An approach for profiling phishing activities. *Comput. Secur.* **2014**, *45*, 27–41. [CrossRef]
64. Basnet, R. Feature Selection for Improved Phishing Detection. In Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Dalian, China, 9–12 June 2012; Springer: Berlin/Heidelberg, Germany, 2012.
65. Li, Y.; Wang, G.; Nie, L.; Wang, Q.; Tan, W. Distance Metric Optimization Driven Convolutional Neural Network for Age-Invariant Face Recognition. *Pattern Recognit.* **2018**, *75*, 51–62. [CrossRef]
66. Chatterjee, M.; Namin, A.S. Deep Reinforcement Learning for Detecting Malicious Websites. *arXiv* **2019**, arXiv:1905.09207.
67. Hamid, I.R.A.; Abawajy, J. Hybrid feature selection for phishing email detection. In Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing, Melbourne, Australia, 24–26 October 2011; Springer: Berlin/Heidelberg, Germany, 2011; Volume 7017, pp. 266–275.
68. Li, T.; Li, J.; Liu, Z.; Li, P.; Jia, C. Differentially private Naive Bayes learning over multiple data sources. *Inf. Sci.* **2018**, *444*, 89–104. [CrossRef]

69. Martin, D.J.; Kifer, D.; Machanavajjhala, A.; Gehrke, J.; Halpern, J.Y. Worst-Case Background Knowledge for Privacy-Preserving Data Publishing. In Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering, Istanbul, Turkey, 15–20 April 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 126–135.
70. Machanavajjhala, A.; Kifer, D.; Gehrke, J. L-Diversity: Privacy Beyond k-Anonymity. In Proceedings of the 22nd International Conference on Data Engineering (ICDE'06), Atlanta, GA, USA, 3–7 April 2006; IEEE: Piscataway, NJ, USA, 2006; Volume 1.
71. Li, N. t -Closeness: Privacy Beyond k-Anonymity and –Diversity. In Proceedings of the IEEE 23rd International Conference on Data Engineering, Istanbul, Turkey, 11–15 April 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 106–115.
72. Sweeney, L. k -ANONYMITY: A Model for Protecting Privacy. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* **2002**, *10*, 1–14. [CrossRef]
73. Gao, C.Z.; Cheng, Q.; He, P.; Susilo, W.; Li, J. Privacy-preserving Naive Bayes classifiers secure against the substitution-then-comparison attack. *Inf. Sci.* **2018**, *444*, 72–88. [CrossRef]
74. Jia, Q.; Guo, L.; Jin, Z.; Fang, Y. Preserving Model Privacy for Machine Learning in Distributed Systems. *IEEE Trans. Parallel Distrib. Syst.* **2018**, *29*, 1808–1822. [CrossRef]
75. Graepel, T.; Lauter, K.; Naehrig, M. ML confidential: Machine learning on encrypted data. In Proceedings of the International Conference on Information Security and Cryptology, Seoul, Korea, 28–30 November 2012; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7839, pp. 1–21.
76. Li, P.; Li, J.; Huang, Z.; Li, T.; Gao, C.Z.; Yiu, S.M.; Chen, K. Multi-key privacy-preserving deep learning in cloud computing. *Future Gener. Comput. Syst.* **2017**, *74*, 76–85. [CrossRef]
77. Li, J.; Wang, L.; Wang, L.; Wang, X.; Huang, Z.; Li, J. Verifiable Chebyshev Maps-Based Chaotic Encryption Schemes with Outsourcing Computations in the Cloud/Fog Scenarios. *Concurr. Comput. Pract. Exp.* **2018**, *31*, e4523. [CrossRef]
78. Shen, J.; Gui, Z.; Ji, S.; Shen, J.; Tan, H.; Tang, Y. Cloud-aided Lightweight Certificateless Authentication Protocol with Anonymity for Wireless Body Area Networks. *J. Netw. Comput. Appl.* **2018**, *106*, 117–123. [CrossRef]
79. Cao, Y.; Zhou, Z.; Sun, X.; Gao, C. Coverless Information Hiding Based on the Molecular Structure Images of Material. *Comput. Mater. Contin.* **2018**, *54*, 197–207.
80. Xiang, C.; Tang, C.; Cai, Y.; Xu, Q. Privacy-preserving face recognition with outsourced computation. *Soft Comput.* **2016**, *20*, 3735–3744. [CrossRef]
81. Xu, J.; Wei, L.; Zhang, Y.; Wang, A.; Zhou, F.; Gao, C.H. Dynamic Fully Homomorphic Encryption-based Merkle Tree for Lightweight Streaming Authenticated Data Structures. *J. Netw. Comput. Appl.* **2018**, *107*, 113–124. [CrossRef]
82. Naercio, M.; Borrego, C.; Pereira, P.; Correia, M. PRIVO: A privacy-preserving opportunistic routing protocol for delay tolerant networks. In Proceedings of the 2017 IFIP Networking Conference (IFIP Networking) and Workshops, Stockholm, Sweden, 12–16 June 2017; IEEE: Piscataway, NJ, USA, 2018; pp. 1–9.
83. Zhang, Q.; Yang, L.T.; Chen, Z. Privacy-Preserving Deep Computation Model on Cloud for Big Data Feature Learning. *IEEE Trans. Comput.* **2016**, *65*, 1351–1362. [CrossRef]
84. Li, P.; Li, T.; Ye, H.; Li, J.; Chen, X.; Xiang, Y. Privacy-preserving machine learning with multiple data providers. *Future Gener. Comput. Syst.* **2018**, *87*, 341–350. [CrossRef]
85. Shokri, R.; Shmatikov, V. Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; ACM: New York, NY, USA, 2015; pp. 1310–1321.
86. Gao, C.Z.; Cheng, Q.; Li, X.; Xia, S.B. Cloud-assisted privacy-preserving profile-matching scheme under multiple keys in the mobile social network. *Clust. Comput.* **2019**, *22*, 1655–1663. [CrossRef]
87. Li, P.; Li, J.; Huang, Z.; Gao, C.Z.; Chen, W.B.; Chen, K. Privacy-preserving outsourced classification in cloud computing. *Clust. Comput.* **2018**, *21*, 277–286. [CrossRef]
88. Zhang, X.; Chen, X.; Wang, J.; Zhan, Z.; Li, J. Verifiable privacy-preserving single-layer perceptron training scheme in cloud computing. *Soft Comput.* **2018**, *22*, 7719–7732. [CrossRef]
89. Vaidya, J.; Shafiq, B.; Basu, A.; Hong, Y. Differentially private naive bayes classification. In Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), Atlanta, GA, USA, 17–20 November 2013; Volume 1, pp. 571–576.

90. Ye, H.; Liu, J.; Wang, W.; Li, P.; Li, T.; Li, J. Secure and efficient outsourcing differential privacy data release scheme in Cyber-physical system. *Future Gener. Comput. Syst.* **2018**. [CrossRef]
91. Zhang, T.; Zhu, Q. Dynamic Differential Privacy for ADMM-Based Distributed Classification Learning. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 172–187. [CrossRef]
92. Zhu, H.; Liu, X.; Lu, R.; Li, H. Efficient and Privacy-Preserving Online Medical Prediagnosis Framework Using Nonlinear SVM. *IEEE J. Biomed. Heal. Inform.* **2017**, *21*, 838–850. [CrossRef] [PubMed]
93. Jagannathan, G.; Wright, R.N. Privacy-Preserving Distributed k -Means Clustering over Arbitrarily Partitioned Data. In Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, Chicago, IL, USA, 21–24 August 2005; ACM: New York, NY, USA, 2005; pp. 593–599.
94. Nazir, A.; Raza, S.; Chuah, C.N. Unveiling facebook: A measurement study of social network-based applications. In Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement, Vouliagmeni, Greece, 20–22 October 2008; ACM: New York, NY, USA, 2008; pp. 43–56.
95. Liu, Z.; Wu, Z.; Li, T.; Li, J.; Shen, C. GMM and CNN Hybrid Method for Short Utterance Speaker Recognition. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3244–3252. [CrossRef]
96. Ntoulas, A.; Najork, M.; Manasse, M.; Fetterly, D. Detecting spam web pages through content analysis. In Proceedings of the 15th International Conference on World Wide Web, Edinburgh, Scotland, 23–26 May 2006; ACM: New York, NY, USA, 2006; p. 83.
97. Zhou, D.; Burges, C.J.C.; Tao, T. Transductive link spam detection. In Proceedings of the Adversarial Information Retrieval on the Web (AIRWeb), Banff, AB, Canada, 8 May 2007; p. 21.
98. Wang, C.; Shen, J.; Liu, Q.; Ren, Y.; Li, T. A Novel Security Scheme based on Instant Encrypted Transmission for Internet-of-Things. *Secur. Commun. Netw.* **2018**. [CrossRef]
99. Webb, S.; Caverlee, J.; Pu, C. Predicting Web Spam with HTTP Session Information. In Proceedings of the CIKM'08 17th ACM Conference on Information and Knowledge Management, Napa Valley, CA, USA, 26–30 October 2008; ACM: New York, NY, USA, 2008; pp. 339–348.
100. Radlinski, F. Addressing malicious noise in clickthrough data. In Proceedings of the Learning to Rank for Information Retrieval Workshop at SIGIR, Ithaca, NY, USA, 2 December 2007.
101. Sydow, M.; Piskorski, J.; Weiss, D.; Castillo, C. *Application of Machine Learning in Combating Web Spam*; IOS Press: Amsterdam, The Netherlands, 2007.
102. Erdélyi, M.; Garzó, A.; Benczúr, A.A. Web Spam Classification: A Few Features Worth More. In Proceedings of the WebQuality'11 2011 Joint WICOW/AIRWeb Workshop on Web Quality, Hyderabad, India, 28 March 2011; ACM: New York, NY, USA, 2011; p. 27.
103. Chen, C.; Zhang, J.; Xie, Y.; Xiang, Y.; Zhou, W.; Hassan, M.M.; Alrubaian, M. A Performance Evaluation of Machine Learning-Based Streaming Spam Tweets Detection. *IEEE Trans. Comput. Soc. Syst.* **2016**, *2*, 65–76. [CrossRef]
104. Singh, A.; Batra, S. Ensemble based spam detection in social IoT using probabilistic data structures. *Future Gener. Comput. Syst.* **2018**, *81*, 359–371. [CrossRef]
105. Wang, X.; Liu, K.; Zhao, J. Detecting Deceptive Review Spam via Attention-Based Neural Networks. In Proceedings of the National CCF Conference on Natural Language Processing and Chinese Computing, Dunhuang, China, 9–14 October 2019; Springer: Berlin/Heidelberg, Germany, 2018; Volume 1, pp. 866–876.
106. Shirani-Mehr, H. SMS Spam Detection using Machine Learning Approach. *Int. J. Inform. Secur. Sci.* **2012**, *2*, 1–4.
107. Kolari, P.; Java, A.; Finin, T.; Oates, T.; Joshi, A. Detecting Spam Blogs: A Machine Learning Approach. In Proceedings of the National Conference on Artificial Intelligence, Boston, MA, USA, 16–20 July 2006; AAAI Press: Cambridge, MA, USA, 2006; Volume 6, pp. 1351–1356.
108. Huang, Z.; Liu, S.; Mao, X.; Chen, K.; Li, J. Insight of the protection for data security under selective opening attacks R. *Inf. Sci. (NY)* **2017**, *412–413*, 223–241. [CrossRef]
109. Munteanu, A.; Fotache, D.; Dospinescu, O. Information Systems Security Risk Assessment: Harmonization with International Accounting Standards. In Proceedings of the 2008 International Conference on Computational Intelligence for Modelling Control & Automation, Vienna, Austria, 10–12 December 2008; IEEE: Piscataway, NJ, USA, 2009; pp. 1111–1117.
110. Asosheh, A.; Dehmoubed, B.; Khani, A. A new quantitative approach for information security risk assessment. In Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology, Dallas, TX, USA, 8–11 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 222–227.

111. Guan, B.-C.; Lo, C.-C.; Wang, P.; Hwang, J.-S. Evaluation of information security related risks of an organization—The application of the multi-criteria decision-making method. In Proceedings of the IEEE 37th Annual 2003 International Carnahan Conference on Security Technology, Taipei, Taiwan, 14–16 October 2003; IEEE: Piscataway, NJ, USA; 2003; pp. 168–175.
112. Munir, R.; Mufti, M.R.; Awan, I.; Hu, Y.F.; Disso, J.P. Detection, mitigation and quantitative security risk assessment of invisible attacks at enterprise network. In Proceedings of the 2015 3rd International Conference on Future Internet of Things and Cloud, Rome, Italy, 24–26 August 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 256–263.
113. Saripalli, P.; Walters, B. QUIRC: A Quantitative Impact and Risk Assessment Framework for Cloud Security. In Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing, Miami, FL, USA, 5–10 July 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 280–288.
114. Ralston, P.A.S.; Graham, J.H.; Hieb, J.L. Cyber security risk assessment for SCADA and DCS networks. *ISA Trans.* **2007**, *46*, 583–594. [CrossRef]
115. de Barros Correia, R.; Pirmez, L.; da Costa Carmo, L.F.R. Evaluating security risks following a compliance perspective. In Proceedings of the 2008 11th IEEE High Assurance Systems Engineering Symposium, Nanjing, China, 3–5 December 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 27–36.
116. Liu, Y.; Ling, J.; Liu, Z.; Shen, J.; Gao, C. Finger Vein Secure Biometric Template Generation Based on Deep Learning. *Soft Comput.* **2018**, *22*, 2257–2265. [CrossRef]
117. Wang, J.; Chaudhury, A.; Rao, H.R. A value-at-risk approach to information security investment. *Inf. Syst. Res.* **2008**, *19*, 106–120. [CrossRef]
118. Wallace, L.; Keil, M.; Rai, A. Understanding software project risk: A cluster analysis. *Inf. Manag.* **2004**, *42*, 115–125. [CrossRef]
119. Eminagaoglu, M. A Qualitative Information Security Risk Assessment Model using Machine Learning Techniques. In Proceedings of the ICT2012 Second International Conference on Advances in Information Technologies and Communication, Amsterdam, The Netherlands, 27–29 June 2018.
120. Jing, Y.; Ahn, G.-J.; Zhao, Z.; Hu, H. RiskMon: Continuous and Automated Risk Assessment of Mobile Applications. In Proceedings of the 4th ACM Conference on Data and Application Security and Privacy, San Antonio, TX, USA, 3–5 March 2014; ACM: New York, NY, USA, 2014; pp. 99–110.
121. Guntamukkala, N.; Dara, R.; Grewal, G. A Machine-Learning Based Approach for Measuring the Completeness of Online Privacy Policies. In Proceedings of the 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 9–11 December 2015; IEEE: Piscataway, NJ, USA, 2016; pp. 289–294.
122. Wei, Q.; De-Zheng, Z. Security Metrics Models and Application with SVM in Information Security Management. In Proceedings of the 2007 International Conference on Machine Learning and Cybernetics, Hong Kong, China, 19–22 August 2007; IEEE: Piscataway, NJ, USA, 2007; Volume 6, pp. 3234–3238.
123. Marwan, M.; Kartit, A.; Ouahmane, H. ScienceDirect Security Enhancement in Healthcare Cloud using Machine Learning. *Procedia Comput. Sci.* **2018**, *127*, 388–397. [CrossRef]
124. Christodorescu, M.; Jha, S.; Seshia, S.A.; Song, D.; Bryant, R.E. Semantics-Aware Malware Detection. In Proceedings of the 2005 IEEE Symposium on Security and Privacy (S&P'05), Oakland, CA, USA, 8–11 May 2005; IEEE: Piscataway, NJ, USA, 2005; pp. 32–46.
125. Alam, S.; Traore, I.; Sogukpinar, I. Annotated Control Flow Graph for Metamorphic Malware Detection. *Comput. J.* **2014**, *58*, 2608–2621. [CrossRef]
126. Sulaiman, A.; Ramamoorthy, K.; Mukkamala, S.; Sung, A.H. Malware Examiner using Disassembled Code (MEDiC). In Proceedings of the Sixth Annual IEEE SMC Information Assurance Workshop, West Point, NY, USA, 15–17 June 2005; IEEE: Piscataway, NJ, USA, 2005; Volume 2005, pp. 428–429.
127. Sung, A.H.; Xu, J.; Chavez, P.; Mukkamala, S. Static Analyzer of Vicious Executables (SAVE). In Proceedings of the 20th Annual Computer Security Applications Conference, Tucson, AZ, USA, 6–10 December 2004; IEEE: Piscataway, NJ, USA, 2004; pp. 326–334.
128. Kreibich, C.; Crowcroft, J. Honeycomb—Creating Intrusion Detection Signatures Using Honeypots. *ACM SIGCOMM Comput. Commun. Rev.* **2004**, *34*, 51–56. [CrossRef]
129. Li, W.J.; Wang, K.; Stolfo, S.J.; Herzog, B. Fileprints: Identifying file types by n-gram analysis. In Proceedings of the Sixth Annual IEEE SMC Information Assurance Workshop, West Point, NY, USA, 15–17 June 2005; IEEE: Piscataway, NJ, USA, 2005; Volume 2005, pp. 64–71.

130. Choudhary, S.P.; Vidyarthi, M.D. A Simple Method for Detection of Metamorphic Malware using Dynamic Analysis and Text Mining. *Procedia Comput. Sci.* **2015**, *54*, 265–270. [CrossRef]
131. Patcha, A.; Park, J.M. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Comput. Netw.* **2007**, *51*, 3448–3470. [CrossRef]
132. Gorji, A.; Abadi, M. Detecting Obfuscated JavaScript Malware Using Sequences of Internal Function Calls. In Proceedings of the 2014 ACM Southeast Regional Conference, Kennesaw, GA, USA, 28–29 March 2014; ACM: New York, NY, USA, 2014; p. 64.
133. Bayer, U.; Comparetti, P.M.; Hlauschek, C.; Kruegel, C.; Kirda, E. Scalable, Behavior-Based Malware Clustering. In Proceedings of the Network and Distributed System Security Symposium, San Diego, CA, USA, 8–11 February 2009; Volume 272, pp. 51–88.
134. Jhaveri, R.H.; Patel, N.; Zhong, Y.; Sangaiah, A. Sensitivity Analysis of an Attack-Pattern Discovery Based Trusted Routing Scheme for Mobile Ad-Hoc Networks in Industrial IoT. *IEEE Access* **2018**, *6*, 20085–20103. [CrossRef]
135. Das, S.; Liu, Y.; Zhang, W.; Chandramohan, M. Semantics-based online malware detection: Towards efficient real-time protection against malware. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 289–302. [CrossRef]
136. Arp, D.; Spreitzenbarth, M.; Hübner, M.; Gascon, H.; Rieck, K. Drebin: Effective and Explainable Detection of Android Malware in Your Pocket. In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 23–26 February 2014.
137. Gavrilut, D.; Cimpoesu, M.; Anton, D.; Ciortuz, L. Malware detection using machine learning. In Proceedings of the 2009 International Multiconference on Computer Science and Information Technology, Mragowo, Poland, 12–14 October 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 735–741.
138. Xu, Z.; Ray, S.; Subramanyan, P.; Malik, S. Malware detection using machine learning based analysis of virtual memory access patterns. In Proceedings of the Conference on Design, Automation & Test in Europe, Lausanne, Switzerland, 27–31 March 2017; European Design and Automation Association: Leuven, Belgium, 2017; pp. 169–174.
139. Rieck, K.; Holz, T.; Willems, C. Learning and Classification of Malware Behavior. In Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Paris, France, 10–11 July 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 108–125.
140. Patel, N.J.; Jhaveri, R.H. Detecting packet dropping nodes using machine learning techniques in Mobile ad-hoc network: A survey. In Proceedings of the 2015 International Conference on Signal Processing and Communication Engineering Systems, Guntur, India, 2–3 January 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 468–472.
141. Thompson, H.H. Why security testing is hard. *IEEE Secur. Priv.* **2003**, *1*, 83–86. [CrossRef]
142. Mitchell, J.C.; Mitchell, M.; Stern, U. Automated Analysis of Cryptographic Protocols Using Murcp. In Proceedings of the 1997 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 4–7 May 1997; IEEE: Piscataway, NJ, USA, 1997; pp. 141–151.
143. Vardhan, A.; Sen, K.; Viswanathan, M.; Agha, G. Learning to verify safety properties. In Proceedings of the International Conference on Formal Engineering Methods, Seattle, WA, USA, 8–12 November 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 274–289.
144. Dupont, P. Incremental regular inference. In Proceedings of the International Colloquium on Grammatical Inference, Tokyo, Japan, 20–22 September 2006; Springer: Berlin/Heidelberg, Germany, 1996; Volume 1147, pp. 222–237.
145. Shu, G.; Lee, D. Testing security properties of protocol implementations—A machine learning based approach. In Proceedings of the 27th International Conference on Distributed Computing Systems, Toronto, ON, Canada, 25–27 June 2007; IEEE: Piscataway, NJ, USA, 2007.
146. Shu, G.; Hsu, Y.; Lee, D. Detecting Communication Protocol Security Flaws by Formal Fuzz Testing and Machine Learning. In Proceedings of the International Conference on Formal Techniques for Networked and Distributed Systems, Tokyo, Japan, 10–13 June 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 299–304.
147. Ghaffarian, S.M.; Shahriari, H.R. Software Vulnerability Analysis and Discovery Using Machine-Learning and Data-Mining Techniques. *ACM Comput. Surv.* **2017**, *50*, 56. [CrossRef]

148. Sánchez-Carmona, A.; Robles, S.; Borrego, C. PrivHab+: A secure geographic routing protocol for DTN. *Comput. Commun.* **2016**, *78*, 56–73. [CrossRef]
149. Carlos, B.; Amadeo, M.; Molinaro, A.; Jhaveri, R.H. Privacy-Preserving Forwarding using Homomorphic Encryption for Information-Centric Wireless Ad Hoc Networks. *IEEE Commun. Lett.* **2019**, *23*, 1708–1711.
150. Lin, Q.; Yan, H.; Huang, Z.; Chen, W.; Shen, J.; Tang, Y. An ID-based linearly homomorphic signature scheme and its application in the blockchain. *IEEE Access* **2018**, *6*, 20632–20640. [CrossRef]
151. Lin, Q.; Li, J.; Huang, Z.; Chen, W.; Shen, J. A short linearly homomorphic proxy signature scheme. *IEEE Access* **2018**, *6*, 12966–12972. [CrossRef]
152. Roli, F.; Biggio, B.; Fumera, G. Pattern Recognition Systems under Attack. In Proceedings of the Iberoamerican Congress on Pattern Recognition, Havana, Cuba, 28–31 October 2019; Springer: Berlin/Heidelberg, Germany, 2013; Volume 3287, pp. 350–357.
153. Šrndić, N.; Laskov, P. Practical evasion of a learning-based classifier: A case study. In Proceedings of the 2014 IEEE Symposium on Security and Privacy, San Jose, CA, USA, 18–21 May 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 197–211.
154. Biggio, B.; Bul, S.R.; Pillai, I.; Mura, M. Poisoning Complete-Linkage Hierarchical Clustering. In Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), Joensuu, Finland, 20–22 August 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 42–52.
155. Tygar, J.D. Adversarial machine learning. *IEEE Internet Comput.* **2011**, *15*, 4–6. [CrossRef]
156. Xiaoa, F.R.H.; Biggiob, B.; Nelsonb, B.; Xiaoa, H.; Eckerta, C.; Department, A. Support Vector Machines Under Adversarial Label Contamination. *Neurocomputing* **2015**, *160*, 53–62. [CrossRef]
157. Barreno, M.; Bartlett, P.L.; Chi, F.J.; Joseph, A.D.; Nelson, B.; Rubinstein, B.I.; Tygar, J.D. Open problems in the security of learning. In Proceedings of the 1st ACM workshop on Workshop on AISec, Alexandria, VA, USA, 27 October 2008; ACM: New York, NY, USA, 2008; p. 19.
158. Chung, S.P.; Mok, A.K. Allergy Attack Against Automatic Signature Generation. In Proceedings of the International Workshop on Recent Advances in Intrusion Detection, Hamburg, Germany, 20–22 September 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 61–80.
159. Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Šrndić, N.; Laskov, P.; Roli, F. Evasion Attacks against Machine Learning at Test Time. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Prague, Czech Republic, 23–27 September 2013; Springer: Berlin/Heidelberg, Germany, 2013; Volume 8190, pp. 387–402.
160. Maiorca, D.; Giacinto, G.; Corona, I. A Pattern Recognition System for Malicious PDF Files Detection Davide. In Proceedings of the International Workshop on Machine Learning and Data Mining in Pattern Recognition, Berlin, Germany, 13–20 July 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 510–524.
161. Goodfellow, J.I.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2015**, arXiv:1412.6572.
162. Porter, F.A.; Chin, E.; Hanna, S.; Song, D.; Wagner, D. Android permissions demystified. In Proceedings of the 18th ACM Conference on Computer and Communications Security, Chicago, IL, USA, 17–21 October 2011; ACM: New York, NY, USA, 2011; pp. 627–638.
163. Huang, X.; Biggio, B.; Brown, G.; Fumera, G.; Eckert, C.; Roli, F. Is feature selection secure against training data poisoning? In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 1689–1698.
164. Sen, C.; Xue, M.; Fan, L.; Hao, S.; Xu, L.; Zhu, H.; Li, B. Automated poisoning attacks and defenses in malware detection systems: An adversarial machine learning approach. *Comput. Secur.* **2018**, *73*, 326–344.
165. Nelson, B.; Barreno, M.; Chi, F.J.; Joseph, A.D.; Rubinstein, B.I.; Saini, U.; Xia, K. Misleading learners: Co-opting your spam filter. In *Machine Learning in Cyber Trust*; Springer: Boston, MA, USA, 2009; pp. 17–51.
166. Florian, T.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; McDaniel, P. Ensemble adversarial training: Attacks and defenses. *arXiv* **2017**, arXiv:1705.07204.
167. Martin, A.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; ACM: New York, NY, USA, 2016; pp. 308–318.

168. Yao, Y.; Ling, S.; Chi, E. Investigation on distributed k-means clustering algorithm of homomorphic encryption. *Comput. Technol. Dev.* **2017**, *27*, 81–85.

169. Metzen, J.H.; Genewein, T.; Fischer, V.; Bischoff, B. On detecting adversarial perturbations. *arXiv* **2017**, arXiv:1702.04267.

170. Kathrin, G.; Papernot, N.; Manoharan, P.; Backes, M.; McDaniel, P. Adversarial perturbations against deep neural networks for malware classification. *arXiv* **2016**, arXiv:1606.04435.