sensors

MDPI

Review

Leveraging Computational Intelligence Techniques for Defensive Deception: A Review, Recent Advances, Open Problems and Future Directions

Pilla Vaishno Mohan [1], Shriniket Dixit [1], Amogh Gyaneshwar [1], Utkarsh Chadha [2], Kathiravan Srinivasan [1] and Jung Taek Seo [3,*]

[1] School of Computer Science and Engineering, Vellore Institute of Technology (VIT), Vellore 632014, India; pillavaishno.mohan2019@vitstudent.ac.in (P.V.M.); shriniket.dixit2019@vitstudent.ac.in (S.D.); amogh.gyaneshwar2019@vitstudent.ac.in (A.G.); kathiravan.srinivasan@vit.ac.in (K.S.)

[2] School of Mechanical Engineering, Vellore Institute of Technology (VIT), Vellore 632014, India; utkarsh.chadha2018@vitstudent.ac.in

[3] Department of Computer Engineering, Gachon University, Seongnam 13120, Korea

* Correspondence: seojt@gachon.ac.kr

Abstract: With information systems worldwide being attacked daily, analogies from traditional warfare are apt, and deception tactics have historically proven effective as both a strategy and a technique for Defense. Defensive Deception includes thinking like an attacker and determining the best strategy to counter common attack strategies. Defensive Deception tactics are beneficial at introducing uncertainty for adversaries, increasing their learning costs, and, as a result, lowering the likelihood of successful attacks. In cybersecurity, honeypots and honeytokens and camouflaging and moving target defense commonly employ Defensive Deception tactics. For a variety of purposes, deceptive and anti-deceptive technologies have been created. However, there is a critical need for a broad, comprehensive and quantitative framework that can help us deploy advanced deception technologies. Computational intelligence provides an appropriate set of tools for creating advanced deception frameworks. Computational intelligence comprises two significant families of artificial intelligence technologies: deep learning and machine learning. These strategies can be used in various situations in Defensive Deception technologies. This survey focuses on Defensive Deception tactics deployed using the help of deep learning and machine learning algorithms. Prior work has yielded insights, lessons, and limitations presented in this study. It culminates with a discussion about future directions, which helps address the important gaps in present Defensive Deception research.

Keywords: defensive deception; machine-learning; deep learning; computational intelligence; honeypots; moving target defense

check for updates

Citation: Mohan, P.V.; Dixit, S.; Gyaneshwar, A.; Chadha, U.; Srinivasan, K.; Seo, J.T. Leveraging Computational Intelligence Techniques for Defensive Deception: A Review, Recent Advances, Open Problems and Future Directions. *Sensors* **2022**, *22*, 2194. https://doi.org/10.3390/s22062194

Academic Editor: Wai Lok Woo

Received: 16 January 2022
Accepted: 9 March 2022
Published: 11 March 2022

1. Introduction

Advanced cyber defenses must provide a quick response against attacker activities in real-time scenarios. They demand clever defense systems that can automatically react to adversarial conduct and evolve with time as the progress of the attack. Before running a defensive action, the AI method utilized by the defensive system should be able to have the foresight and analyze the pattern of an attacker to take appropriate defensive measures. Adaptive or active cyber security, in which a system plans and uses defense techniques automatically in response to an identified suspicious activity without human intervention, is growing rapidly, but it has not yet been extensively adopted.

Cyber Deception is one of the major techniques in cyber defense research. In comparison to standard security safeguards, deception-based systems operate fundamentally differently [1–6]. Traditional security measures are employed in response to the actions of an attacker, detecting or preventing them, whereas deception-based measures are used in anticipation of such actions, manipulating attackers' perceptions and thus inducing adversaries to take decisions that are advantageous to systems which the adversary is targeting.

Deception is especially significant in military-style attacks that are time sensitive, such as those carried out by cyber terrorists, where simply postponing the attack with the help of deceptions could be crucial until a permanent defense is developed [7]. Both insider and outsider attacks can be prevented using Deception. These days machine learning has emerged as an effective technology that provides us with a wide range of applications ranging from recognition of patterns, image identification, image, and video processing, making predictions, virus or malware detection, autonomous driving, and other application scenarios [8–39]. The advantages of machine learning algorithms can be extended for deploying Defensive Deception frameworks [40–89]. Deception has been employed in honeypots, which are legal traps and honeynets (honeypot networks), as a defensive tool for information systems to keep attackers occupied [90–123]. Honeypots are systems that exist solely to promote attacks in order to collect data. Interconnected honeypot networks are known as honeynets. Some honeypots employ deceptions such as phony files to entice attackers to stay away from actual resources for a while. Moving target defense, a type of deception technology, makes an attacker's work more difficult by adding unpredictability to the attack area and changing information quickly. By incorporating falsehoods and obscuring real facts, Deception can add a new level of ambiguity. It can immensely affect the decision-making of an attacker, forcing them to squander time and effort.

Furthermore, a defense can utilize cyber Deception to give the attacker the wrong impression. This erroneous notion can generate ripple effects throughout the cyber death chain, disrupting several attacks over time. There are two major promising paths for developing Defensive Deception tactics in this literature. First, attacker and defender strategies have been commonly described using machine learning, with the defender employing Defensive Deception strategies to confuse or mislead attackers into choosing suboptimal or inferior strategies. Second, this article discusses deep learning-based Defensive Deception approaches implemented in recent cyber security advancements. The article then progresses with various taxonomies used in Deception and their description. Finally, the article concludes with future research directions and solutions for the same.

1.1. Contribution of this Survey

Our contribution can be summarized as follows:

- This is the first survey that briefly discusses the application of various Machine learning and deep learning methods in the implementation of Defensive Deception and its technologies.
- Discussion on new techniques in Defensive Deception such as Genetic Algorithms, Multi (Intelligent) Agents, DBN, SOM, etc., along with the traditional Computational Intelligence techniques such as KNN, Random Forest, ANN, DNN, etc.
- Detailed tabular summary of works on Machine Learning and Deep Learning Techniques in Defensive Deception are included. The summary provides the model, key contributions, and limitations for the same.
- A brief description of various methods to implement Defensive Deception has been provided. This includes Perturbation, Moving Target Defense, Obfuscation, Mixing, Honey-x, and Attacker Engagement.
- Classification of several deception categories and commonly used datasets have been mentioned.
- Finally, the paper describes various open challenges present in Defensive Deception and future research directions for further improvements in this field.

Table 1 presents the current review articles of the CI-enabled techniques in defensive deception.

Table 1. Review articles of the CI-enabled techniques in Defensive Deception (✓: Yes, ×: No).

Ref.	Year	No. of Articles	Brief on Focus (One-Sentence Summary)	CI-Enabled Techniques		Open Challenges	Future Directions
				Machine Learning	Deep Learning		
[39]	2011	28	A Review of Classification Approaches Using Support Vector Machine in Intrusion Detection	✓	×	×	✓
[12]	2012	191	Review article on Nature-Inspired Techniques in the Context of Fraud Detection	✓	×	×	✓
[24]	2012	72	Review article on employment of Data Mining Techniques for financial frauds detection.	✓	×	✓	✓
[18]	2013	62	A review article on Computational Intelligence Models for Insurance Fraud Detection	✓	×	×	✓
[4]	2015	91	A review on application of AI techniques for combatting cybercrime	✓	×	✓	✓
[1]	2018	77	A survey of Artificial Intelligence in Cyber security	✓	✓	×	✓
[16]	2018	41	Review article on employment of machine learning techniques for financial frauds detection.	✓	×	×	✓
[29]	2018	111	A Survey article on Cyber Defensive Techniques employed with the help of Machine Learning algorithms	✓	×	✓	✓
[28]	2019	380	A review of defensive tools and technologies employed in cyberspace	✓	×	✓	✓
[31]	2019	173	A Survey on implementation of adaptive technologies in Moving Target Defense	✓	×	✓	✓
[32]	2020	65	A review article on the implantation of Artificial Intelligence technologies in Electronic Warfare Systems and their applications	✓	✓	×	✓
[34]	2020	145	A Survey article on the implementation of AI, machine learning, and blockchain technology in IoT security	✓	×	✓	✓
[6]	2020	75	A review of deception technologies used in cyber security and user privacy.	✓	×	✓	✓
[26]	2020	83	Review article on AI and machine learning for cybersecurity	✓	✓	×	✓
[30]	2020	175	A Survey article on Moving Target Defenses in order to implement Network Security	✓	×	✓	✓
[25]	2021	187	A Review of Defensive Deception techniques Employed with the help of Game Theory and Machine Learning.	✓	✓	×	✓
Our Review	2022	77	Our review has briefly described various prominent ML and DL models and their use in Deception Technologies.	✓	✓	✓	✓

1.2. Survey Methodology

1.2.1. Search Strategy and Literature Sources

Databases such as ACM Digital, IEEE, Science Direct, etc., were used to find relevant articles. The keywords utilized were: Defense Deception, Fraud Detection, Cyber Defensive Systems, etc., alongside some other keywords relating to the possible fraud types. A total of 1138 non-duplicate articles were found from these databases initially.

1.2.2. Inclusion Criteria

The articles included were based on their relevance. The articles were included based on the novelty of this review's topic and appropriate language, and only English articles were included.

1.2.3. Elimination Criteria

The eliminations of the articles are based on abstract screening, then based on full text and data extraction in the next iteration. The articles were eliminated due to lack of relevance, duplicate articles, articles not in the English language, or poorly written manuscripts.

1.2.4. Results

There were 1138 articles shortlisted from various databases, and after inclusion/exclusion criteria, 77 articles were included for the review, which kept direct relevance with the defense deception; Figure 1 shows the PRISMA implementation for the same.

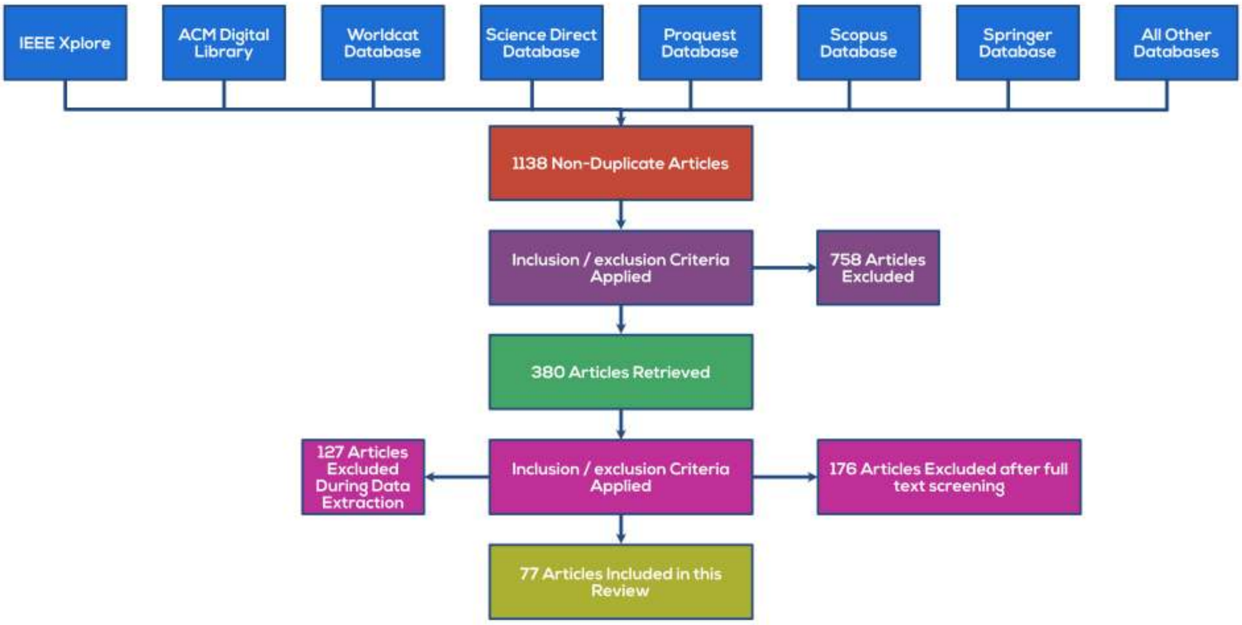
IEEE Xplore
ACM Digital Library
Worldcat Database
Science Direct Database
Proquest Database
Scopus Database
Springer Database
All Other Databases
1138 Non-Duplicate Articles
Inclusion / exclusion Criteria Applied
758 Articles Excluded
380 Articles Retrieved
127 Articles Excluded During Data Extraction
Inclusion / exclusion Criteria Applied
176 Articles Excluded after full text screening
77 Articles Included in this Review

Figure 1. PRISMA flow diagram for the selection process of the research articles used in this review.

1.3. Survey Structure

This survey is prepared by referencing more than 70 research articles. Section 1 of this article consists of a brief overview. The selection process involved for the referenced articles is discussed, and a brief comparison has been performed for the various surveys involved. Section 2 discusses various CI-enabled techniques applied in Defensive Deception technologies. This section is divided into two major subsections. The first subsection includes a brief description of various machine learning algorithms applied in Defensive Deception technologies. The second subsection consists of deep learning algorithms and various applications to implement Defensive Deception technologies. In Section 3, we have described frequently used datasets in our survey, the various Defensive Deception taxonomies used and their implementation in real-world Defensive Deception technologies. Section 4 includes various open problems present in Defensive Deception and a brief description of future research directions. Finally, Section 5 includes the conclusion of this article, followed by the list of references at the end.

2. CI-Enabled Techniques Used in Defensive Deception

Computational intelligence consists of two substantial branches of artificial intelligence technologies: deep learning and machine learning. These methods can have a wide range of applications in Defensive Deception technologies. By merging autonomic computing and cyber Deception, we can obtain an early defender advantage and counter attacker behaviors through automatic adaptation. Article [8] proposes implementing the adaptive deception framework, which involves a tiny network consisting of two Windows 7 client computers and a database server. One hundred runs were performed for four different scenarios where the attacker tried to access this network. For the first control condition, no obstacles were present. As a result, all 100 runs were a success for the attacker, with an average run time of 250.05 s. For static decoys condition, decoys are pre-configured and pre-deployed. The attacker only succeeded 42 times in this situation, was unable to exploit and pivot 19 times, and failed to exfiltrate the database 39 times. The average time to success was 261.80 s, which was somewhat higher than the control average. In the delay condition, decoys are a pre-deployed but adaptive deception system with delay. In this condition, the attacker was successful 40 times, had 27 exfiltration failures and 23 pivot failures. The average successful run took 630.23 s. In the deny condition, decoys are a pre-deployed and adaptive deception system with denying. The attacker was successful 11 times, had 78 pivot failures and 11 exfiltrating failures. The average successful run time was approx. 256.64 s [8]. This article showed how the autonomous deception framework increased the attacker runtime by 175% and reduced the successful runs by 89%, resulting in an optimal defense strategy.

2.1. The Evolution and Overview of AI-Enabled Techniques

In the early stages of AI technology, we majorly tackled cyberspace threats using machine learning (ML) techniques. Although machine learning is extremely strong, it relies majorly on feature extraction. Researchers began studying deep neural networks, often referred to as Deep learning, a sub-domain of machine learning, in response to glaring problems in classical ML. Traditional ML and DL vary in that DL methods can be used directly for training and testing the original data without having to Remove or change their characteristics [1]. In the last few years, DL algorithms have shown a performance improvement of about 20–30% in image processing, natural language processing, and text recognition, and had a significant impact on the development of AI and have a major application in Defensive Deception technologies [1].

2.2. Machine Learning Techniques

ML algorithms are majorly used in AI systems to extract models using raw data. Finding ML solutions includes four major steps.

1. Extracting the features.
2. Selecting an appropriate machine learning algorithm.
3. After evaluating different algorithms and adjusting parameters, training the models, and selecting the model with the best performance.
4. Making predictions for the unknown data with the help of the trained model [1].

The most frequent supervised approaches are those based on supervised machine learning algorithms, which collect large datasets and classify an account as either person or bot. Machine learning includes several strategies that can improve the accuracy of protection. When used effectively, ML powerful algorithms create a learning environment for systems, accomplishing tasks such as spotting known/unknown malicious attacks [2,3,6]. Figure 2 lists all the ML techniques utilized in Defense Deception.

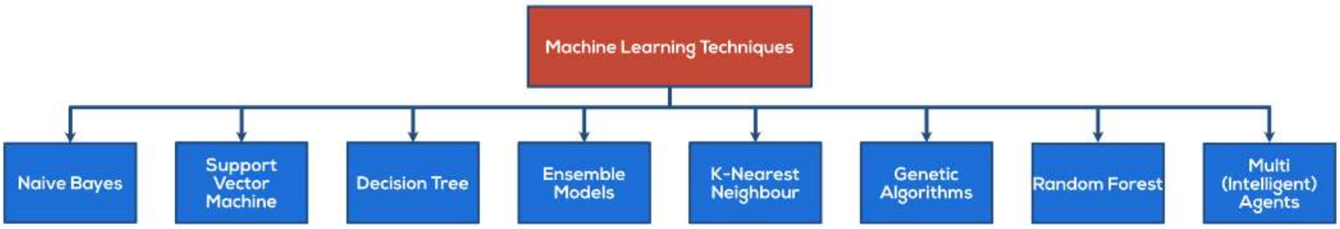

Figure 2. Current machine learning models in defensive deception—nomenclature.

2.2.1. Naïve Bayes

The Bayes conditional probability rule is used in Naïve Bayes (NB), a classification tool. Every attribute along with the class label is treated as a random variable, then the naive Bayes algorithm selects a class for the newly fetched observation that maximizes its probability following the values of the various attributes, provided that the attributes are independent [1,16]. Although Naïve Bayes classifiers weaken when the features are derived from dependent events, they are extensively used because they assume a naive assumption (that every feature is derived from independent events) and can still produce acceptable results [5]. Naïve Bayes analysis works well for deception planners, taking the suitability of Deception into account and planning the type of Deception that needs to be deployed [11,30,113,120]. Naïve Bayes classifiers are widely used for email spam detection and network intrusion detection, which involves deceiving in order to cause harm to the system [12]. The probabilities of the three hypotheses, "network is down," "bugs in the system," and "deception," can be calculated when a download attempt has been made as well as when an attempted modification has occurred, using a Naive Bayes approach. As a result, despite its low initial likelihood, the contradictory signs make Deception more feasible than the other hypotheses [23]. Naive Bayes is a useful categorization method that is simple to understand, and it is especially useful when the inputs include many dimensions [24].

2.2.2. Decision Tree

The decision tree method is majorly used for extracting a set of inferences by analyzing the derived rules from a couple of training datasets or samples. The decision tree first finds a feature that can categorize the data samples iteratively. After each division, rules are generated for each part of the category. It, in turn, results in a tree-like structure. The process continues until only one class is identified for the data samples [1,5,16]. Because it reveals the result of choice based on feature values, the methodology can be extensively used for detecting cybersecurity issues. This can be achieved by classifying the observed cybersecurity events or occurrences as either being legitimate or an attack.

Furthermore, we can classify data in real time once the tree is defined [5]. We can deceive adversaries by employing probabilistic decision trees to make decisions. These trees can be built using grammar which specifies how a system should react in case of security threats. This technique can be built with the help of a historical dataset (playback) and a network simulation in real time [13]. Machine-learning-based techniques such as the ID3, CART and C4.5 can be used to grow these trees. Leaves indicate predictions, while branches represent feature combinations. Credit cards, auto insurance fraud and corporate fraud involve decision trees. The classification and regression trees, also known as the CART technique, are prominently used to detect and predict the impact of false financial statements [24]. When we have a group of honeypots (a honeynet), rather than just one, a decision tree is more useful to decide which honeypot configuration is best to deploy according to the given scenario. We can also independently test other techniques to determine how well they work and what risks they entail. This can be achieved by calculating the average benefit for several honeypots and honeynet layouts, and the one with the highest average benefit can be chosen [62,63,121].

2.2.3. k-Nearest Neighbour

The k-Nearest Neighbour, commonly known as the k-NN approach, learns with the help of data samples to build classes or clusters. The proposal for k-NN was made as a non-parametric form of pattern analysis [73] which can be used for determining the fraction of data samples in a neighborhood that can produce a consistent probability estimate. To form clusters, the neighborhood is first established with the help of a k-number of data samples, usually based on a distance measure (Euclidian distance, Manhattan distance, etc.). When a dataset sample is newly introduced, it is grouped with one of the clusters based on the votes of all k neighbors. Even for tiny values of k, this strategy is computationally challenging. However, it is appealing for intrusion-detection systems to learn from new traffic patterns and detect zero-day attacks, which are attacks that are not yet known to the vendor or general public [5]. After the attack has been detected correctly, we can deploy appropriate deception decoys to protect the resources.

2.2.4. Random Forest

Random forest works by creating various decision trees from an arbitrarily selected subset of training samples and variables. A random forest classifier is simple to learn and use and quick to test. This learning method is well known for handling nonlinearity and outliers and compatibility with big datasets simultaneous training. A strategy based on decreasing entropy once a dataset is split into separate qualities is known as the information gain feature [3]. A list of 13,884 SQL statements was utilized in the dataset, compiled from multiple sources. 12,881 are malicious (SQL Injections), while 1003 are legitimate. They removed extreme values and outliers during data pre-processing. When 10-fold cross-validation is applied to the dataset, it has an accuracy of 99.1% for SQLI prediction [27]. They used Random Forest to classify the material polluters and then used conventional boosting and bagging and alternative feature group combinations to improve the findings. The authors were able to obtain a higher rate of social adversary collection with the help of a random forest model and, as a result, were able to improve the social honeypots. The upgraded Honeypot collected social enemies 26 times faster than an unaltered social honeypot [25] based on a random forest classifier evaluation.

2.2.5. Support Vector Machine

In order to perform machine learning tasks, Support Vector Machine—commonly known as SVM learning—is a prominent and widely used method. Support vector machine falls under supervised machine learning technologies for categorizing data. This division methodology employs a series of training examples, each of which is classified into one of two groups. After that, the SVM is used to create a model that can predict if a new sample instance belongs to one of two categories using a separating plane. This categorization method aids systems in providing tiny sample sets with improved learning capabilities. The SVM approach can be widely applied in network intrusion detection, online page identification, and facial recognition applications. When used in intrusion detection systems, SVM offers benefits which include high training and decision rates, insensitivity to input data dimension, and constant correction of multiple parameters with a boost in training data, enhancing the system's ability to self-learn [1,3,5,16,18,122,123]. Email spam detection is a successful implementation performed using SVMs [12]. Support Vector Machines can outperform neural network models and cluster and classify outliers using a higher dimensional feature space obtained from the training dataset [40,41]. To find malicious profiles and obtain data from these profiles, the authors used feature-based techniques and honeypot strategies and then evaluated the data using Support Vector Machines (SVM) and other machine learning algorithms [42]. They coined the term "active honeypots," which are Twitter accounts that can catch as many as ten new spammers in a single day. They used Twitter to find 1814 accounts and looked at the essential characteristics of active honeypots. Furthermore, the authors investigated the impact of unbalanced datasets on detection accuracy for various ML methods using a suite of ML techniques, including SVM [28,39,43].

2.2.6. Ensemble Models

Ensemble approaches are useful for security use during the testing or inferring phase. A vast body of work aimed at designing Moving Target Defense systems, commonly known as MTDs, highlights the security benefits while ignoring the performance drawbacks. It is worth noting that the performance impact of MTDs might occur for various reasons. Each MTD ensemble system configuration has an efficiency cost attached, and switching to a high-cost arrangement influences performance [30]. Ensemble models can demonstrate the MTD security benefits by contrasting them with an unaltered system configuration [54,55,66,86,123].

2.2.7. Genetic Algorithms

Even though game-theoretic MTD approaches are the most popular, other techniques such as genetic algorithm is another viable option. Genetic algorithms (GAs) are frequently employed to maximize solutions' optimality. Furthermore, in some fully dispersed setups, ensuring a centralized organization to make MTD decisions based on GAs could be impossible [31].

2.2.8. Multi (Intelligent) Agents

These agents provide proactive cyber-defense techniques such as gathering data, assessing security, monitoring network state, attack detection and countermeasures, malefactor deception, etc. Machine learning techniques applied to the usual interaction between agents in a multi-agent system, for example, can result in coordinated actions and plans emerging on their own Multi-agent system (MAS). Agents are expected to gather information from various sources, use partial knowledge, predict the intentions and behaviors associated with other agents, make decisions according to the actions of other agents and attempt to deceive opposing team agents [9,10].

Table 2 provides an executive summary of the machine learning research works in Defensive Deception.

Table 2. A summary of works on machine learning techniques in defensive deception.

Ref.	Deception-Category	Machine Learning Approaches Used	Key Contribution	Limitations
[25]	Honeypots, honey webs, honeynets, honey flies, HMAC, Moving target defense, obfuscation.	K-Means, Support Vector Machine, Hierarchical Grouping, Expectation-Maximization (EM), Bayesian Network (Bayes Net), Decision Tree (DT), Naïve-Bayes Algorithm, C4.5 Algorithm.	This work is primarily concerned with reviewing game-theoretic and machine learning-based Defensive Deception approaches and addressing the findings, limits, and lessons learned from this comprehensive study.	Various deep learning and machine learning approaches such as genetic algorithms, Ensemble Models, Self-organising maps, etc., were not taken into account for Deception.
[30]	Moving target defense	Ensemble model used	This research first classified various Moving Target Defenses according to the surfaces on which these defenses operate. Secondly, they talked about how these MTDs can be put into effect. It discussed the various measures used to assess the effectiveness of MTDs and drew attention towards domains of network security in which the scope of the construction of MTDs is yet to be explored.	The survey did not consider better machine learning and deep learning approaches to implement moving target defenses.
[65]	Honeypot	C4.5, Decision Tree, Naive-Bayes and Bayes Net.	They employed a machine learning method to predict the most vulnerable and easily attackable host in an SDN (Software Defined Networking) network. The security rules for the SDN controller can be developed using the prediction output of machine learning algorithms to prevent unauthorized user access. The experiments revealed that machine learning techniques could enhance security rules for SDN controllers by properly anticipating potential susceptible hosts. The Bayesian Network achieved about 91.68 percent of average prediction accuracy.	New machine learning approaches such as neutrosophic sets were not taken into consideration.

Table 2. *Cont.*

Ref.	Deception-Category	Machine Learning Approaches Used	Key Contribution	Limitations
[66]	honeypots	Logistic Regression, SVM, KNN, Naive Bayes, ensemble-based models, Random Forest with Gini, and Extra Tree classifiers with Gini.	They demonstrated that fraudulent clicks on Instagram might boost the popularity index of posts through a variety of tactics with their research. They used honeypots and botnets to launch assaults and collect data from various real and false accounts, such as clicks on various posts. Experimental data show that LR is the most accurate predictor among all the single-based approaches, and among all ensemble-based methods, Random Forest is the best.	They did not consider various other approaches such as hybrid learning models, ANN, etc., in order to validate whether a view is legitimate or fake based on the chosen criteria.
[23]	Obfuscation, Honeypot	Naïve Bayes	They methodically cataloged and ranked the available information system deception options, both offensively and defensively. Then they thought about how Defensive Deceptions could be packaged into "generic explanations" that an attacker would find more persuasive than individual refusals to accept directives.	Latest and better machine learning approaches were not used.
[13]	Obfuscation, Honeypot	Decision Tree	A unique deception strategy was developed for network defenses that achieve reactive unpredictability by combining security postures and probabilistic decision trees. They developed a new grammar for decision-tree that allows analysts to specify and identify potential responses based on warnings, mission processes, security postures, and various asset conditions. A real-time simulation based on an organization and its activities and a historical dataset were used to implement, demonstrate, and assess our technique.	A probabilistic decision system can learn optimal decision tree order execution and security postures. Trees that are manually or automatically generated should potentially be improved to boost speed, especially as they grow larger. Attacks are not learned in the current implementation.
[31]	Moving target defense	Genetic algorithm	They conducted a thorough study of MTD techniques, their core classifications, important design features, frequent attack behaviors addressed by existing MTD implementations. The literature also explored various application fields for the MTD techniques.	This article only briefly investigated the relationship between MTD and other defense systems. There has been little research that looks into the influence of MTD on minimizing attacks after the reconnaissance stage. There has not been much research into the best way to use numerous hybrid MTD approaches. Existing MTD methodologies have limitations in monitoring several parameters of a system's quality.
[42]	Honeypots	Support vector machine	They described the creation of a novel honeypot-based social bot in order to detect malicious profiles present in social networking groups. Their overall study goal is to look at techniques and propose effective solutions to automatically recognize and filter the profiles of harmful people who target social networking platforms. In order to attract fraudulent accounts, their strategy employs social honeypot personas.	The SVM algorithm used in this article is not suitable for large datasets. It does not perform very well when the dataset has more noise which is the usual case for Twitter accounts.

Table 2. *Cont.*

Ref.	Deception-Category	Machine Learning Approaches Used	Key Contribution	Limitations
[61]	Perturbation	Artificial neural network	They demonstrated how ANN might be used to modestly adjust the output probabilities by perturbing the final activation layer of the model. The opponent is forced to ignore the class probabilities, making it necessary to use more queries before successfully performing an attack.	Other machine learning and deep learning approaches were not considered for implementing the system.
[63]	Honeypot	Decision tree	A decision tree is more useful when we have a honeynet rather than just one. Then we may independently test other techniques to determine how well they work and what risks they entail. This is achieved by calculating the average benefit for several honeypots and honeynet layouts, and the one with the highest average benefit is chosen.	Other machine learning algorithms were not used to examine the various scenarios generated by honeynet.

2.3. Deep Learning Models

Figure 3 shows all the current deep learning models in defensive deception, which will be explained in this section below.

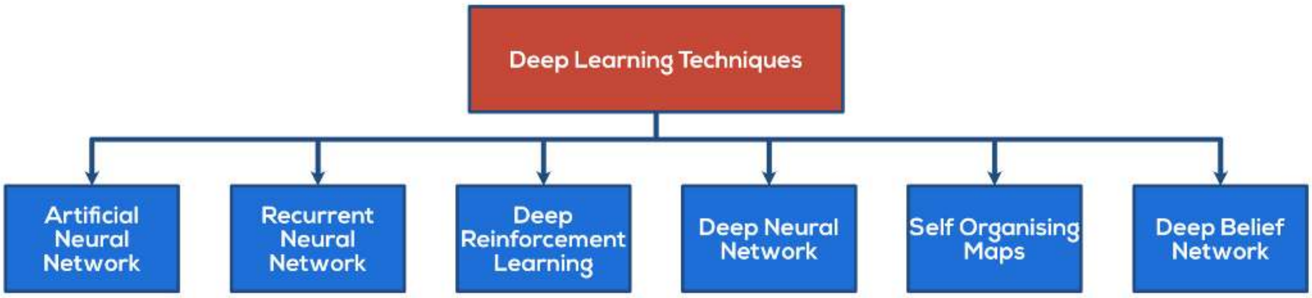

Figure 3. Current deep learning models in defensive deception—nomenclature.

2.3.1. Artificial Neural Network

Artificial Neural Networks (ANN) have computational abilities that help them simulate functional and structural aspects of neurons present in biological systems. They are capable of performing parrel processing of information and high-speed decision-making. These properties make them suitable for attack pattern recognition, classification, and response selection. These can not only be used for IDPS (Intrusion detection and prevention system), but there are also proposals for their application in DOS, malware, worm, and spam detection systems along with forensic investigations [4]. An ANN application was employed in a cybersecurity investigation that used the Cascade Correlation Neural Network (CCNN), which adds additional hidden units to the currently present hidden layers under the algorithm. In this study, the CCNN allows the network to analyze and learn from traffic patterns generated by desktop platform to detect port scanning of mobile networks without requiring the entire network to be retrained with the original data [5]. Another advantage of ANN is that it can detect zero-day attacks due to its ability to learn from previous instances. For example, labeled training data, including traffic patterns generated from DoS attack instances, were fed into ANNs, after which the neurons were able to detect hidden DoS attacks [5]. Users can utilize ANN in cloud-based models to get useful class probability information while reducing the chances of an adversary stealing the model. The last activation layer of the model can be perturbed using ANN, slightly modifying the output probabilities. The adversary is forced to ignore the class probabilities, making it necessary to use more queries before successfully performing an attack. The evaluation demonstrates that such a defense can reduce the stolen model's accuracy by at least 20%, or 64 times increase in the number of queries necessary for an adversary, all with a small impact on the protected model's accuracy [61].

2.3.2. Recurrent Neural Networks

Unlike typical feed-forward neural networks, Recurrent neural networks use directional loops to process sequence data and manage contextual correlation among inputs [1]. RNN (Recurrent Neural Networks) can handle time-series data and raw input feature values and capture data involving changes over time. Within five seconds of running the report, a collection of RNNs can assess whether traffic is malicious or benign with a 97 percent accuracy rate [3]. Using the KDD CUP 1999 dataset, they used Recurrent Neural Networks (RNNs) to identify intrusions and achieved a full detection rate with only a 2.3 percent false alarm rate [46]. The deception jammers were integrated into legitimate systems to make them harder to recognize and more desirable targets. To improve the fidelity of the additional decoy devices to the actual system, three properties are maintained: a protocol, parameters, and logic of the deployed false devices. The authors used a dataset collected over a year to train a recurrent neural network (RNN) to understand such system properties. RNN (Recurrent Neural Networks) was also utilized to generate fake devices that looked real, based on a year of observations of device behavior in a CPS [25].

2.3.3. Deep Neural Network

DNN (Deep Neural Networks) consists of neural networks with a big number of disguised layers [15]. DNNs are a subset of ANNs. Multiple hidden layers are employed in DNNs, allowing various algorithms to analyze variables that would otherwise go unnoticed if only a single layer were used [5]. DNN outperforms neural networks in terms of capacity to fit complex mappings. DNN collects features layer by layer, combining low-level and high-level features in the process. Deep Belief Networks, Stacked Autoencoder and Deep Convolution Neural Networks (DCNN) are three regularly utilized DNN models [32,51,52]. The paper's authors introduced the MTD framework for DNNs, which improves their security and resilience against adversarial assaults. The ideal switching strategy for MT Deep is the Stackelberg equilibrium of the game, which reduces misclassification while retaining excellent classification accuracy for genuine system users on neutrally produced images [31,53].

2.3.4. Deep Belief Network

It is a probability generation model which uses several limited Boltzmann layers to generate probabilities. DNN collects features layer by layer, combining both high-level and low-level features in the process. DBN, SAE, and DCNN are the three most often utilized DNN models. It is a probabilistic unsupervised deep learning algorithm. DBNs comprise layers of Restricted Boltzmann Machines [32,52], followed by a feed-forward network for the fine-tuning step [32,52]. Combining a Deep Belief Network and a probabilistic neural network can result in a novel intrusion detection system that can be used to configure deception measures. The original data were turned into low-dimensional data in this method, then DBN (a nonlinear learning algorithm) identified the main properties from the original data. A particle swarm optimization technique optimized the number of hidden-layer nodes per layer. The low-dimensional data were then classified using a PNN (probabilistic neural network). The "KDD CUP 1999" dataset revealed that this technique outperforms classic PNN, PCA-PNN, and original DBN-PNN without simplification [1]. After the intrusions are detected, appropriate deception decoys can be deployed for protection.

2.3.5. Deep Reinforcement Learning

Deep Reinforcement Learning is a hybrid of reinforcement and deep learning. Reinforcement learning is a branch of machine learning that involves executing appropriate action to maximize the reward in a given situation. It is used to determine the best possible conduct or path to pursue in a given situation [32,56]. Deep Reinforcement Learning (DRL) estimates difficult functions with high-dimensional inputs using a neural network. The addition of deep learning to traditional RL approaches improves the ability to capture the huge scale of numerous Internet-connected systems, such as mobile networks and IoT devices [15]. DRL could develop a low-dimensional version of high-dimensional data, which is quite compact in nature. DNN is a powerful complement, allowing it to be used for the cyber security of vast networked systems [15]. There has been a recent surge of research on using RL to select an adaptive configuration strategy to maximize the impact of MTD, with particular emphasis on the dynamic environment, reduced resource consumption, usability, partially observable environments, and multi-agent scenarios that include both the system's characteristics and the adversary's observed activities. Through a compromise between usability and security, we can investigate the potential of RL in reconfiguring defenses [15]. Creating hybrid Defensive Deception tactics that incorporate machine learning and game theory: Other protection mechanisms have been considered using machine learning-based game-theoretic techniques. Other researchers have developed hybrid approaches that combine reinforcement learning and game theory, using RL as one of the most important parts of the machine learning technique. As in other attack–defense games, Reinforcement Learning's reward functions can be utilized to create players' utility functions and allow an RL agent to determine an ideal strategy [25,57,58].

2.3.6. Self Organizing Maps

A self-organizing map (SOM) is a neural network that uses unsupervised learning instead of supervised learning [16]. Self-organizing maps (SOMs) are a popular data visualization tool enabling the representation of a multi-dimensional dataset on a two-dimensional or three-dimensional map. In other words, they help us to perform dimensionality reduction [5,17]. The approach learns by searching for correlations in data samples. Adjacent data samples have more in common with each other than samples further apart, resulting in data clustering and a map as an output. Because SOMs are computationally intensive, they are unsuitable for real-time systems. Their main advantage is their capacity to visualize data, which is important for detecting network irregularities and understanding the best deception decoy deployed [5]. Table 3 summarizes works on Deep Learning Models in Defensive Deception.

Table 3. A summary of works on Deep Learning Models in Defensive Deception.

Ref.	Deception-Category	Deep Learning Models Used	Key Contribution	Limitations
[71]	Money related deception		There is also a new term, Honeyfile, used in this article. Honeyfiles are also used to create confusion and apprehension about the value and location of sensitive data. This method is based on humans' inability to discern between authentic and bogus information.	There comes a time when cyber security is being scrutinized by the public due to an increasing number of occurrences, even though only a fraction of these instances can be traced back to particular individuals or groups of Blackhats.
[36]	Honeypots, Perturbation	Online Adaptive Metric Learning		Because honeypots are completely "fake systems," there are a variety of methods available to determine whether the present system is a honeypot or not. They are built with this underlying restriction in mind.

Table 3. *Cont.*

Ref.	Deception-Category	Deep Learning Models Used	Key Contribution	Limitations
[68]	Honeypots	Recurrent neural network	This study describes a distributed infrastructure capable of deploying decoys across different network segments and managing their physical world perspectives. This solution's prototype implementation and use case for a boiler model are only two examples of how this new methodology could be used.	To better understand and improve the situation, more research is required. Betterment of fidelity of decoys by generating vendor/product-specific characteristics that include things such as protocols used, ports used, and register point settings.
[53]	Moving target defense, perturbation	Deep neural and deep convolution neural network	They offered MT Deep, a cybersecurity architecture influenced by MTD, as a security service to improve the SAFETY of Deep Neural Network-based classification systems in this study (DNNs). To design the interaction among both MT Deep and users, they used a Bayesian Stackelberg Game. The equilibrium provides the best alternative to the multi-objective problem of lowering misclassified rates on adversarial changed visuals while retaining better classification accuracy on photos images that have not been disturbed.	This article did not examine other neural networks, such as RNN, self-organizing maps, etc.
[15]	Moving target defense	Deep neural network, deep convolution network, and deep reinforcement learning.	The authors have labeled the architecture of RL-based CRM (RL-CRM) according to the types of vulnerabilities it attempts to address. They have shown that the RL-CRM can set up moving target defense, engage attackers for reconnaissance, and lead human attention to mitigate visual weaknesses adaptively and autonomously. Their research revealed that posture-related defense technologies are well-developed, but mitigation options for information-related and human-induced vulnerabilities are still in the early stages of development.	The first hurdle in the learning process is to deal with system and performance limits. Many system limits exist in cyber systems that must be explicitly considered. The improvement of learning speed is a second difficulty. CRM's (Cyber-Resilient Mechanism) purpose is to restore the cyber system following an attack. Fast learning would allow for a more rapid and resilient response to an attack. Dealing with the non-stationarity of cyber systems is the third difficulty. The environment is assumed to be stationary and ergodic in traditional RL algorithms.
[69]	Honeypot, obfuscation	Deep neural network, deep reinforcement learning	They first introduced SRG (System Risk Graph), a precise adversarial model for extracting specific dangers and internet treatments, such as vulnerabilities in the software and virtualization layers. The adversarial model is updated based on the existing condition system. They proposed a deception rate, which is a statistical parameter for evaluating the efficiency of the deployment method based on SRG. Second, they tweaked a DRL algorithm to develop an adjustable decoy deployment strategy for a rapidly changing internet. Finally, they compared the proposed methodology to existing research using simulations.	This article did not analyze other neural networks such as recurrent neural networks, convolution neural networks, etc.
[70]	Honeypot, obfuscation	Deep neural network, Online Adaptive Metric Learning	A machine learning-based framework for evaluating cyber deception defenses with minimum human participation is developed and implemented. This avoids the problems that come with fraudulent research. Humans, ensuring that automated evaluations are as effective as possible, must be completed prior to human study. Only after this can the next step begin.	They were unable to apply labels to previously unknown categories automatically.

Table 3. *Cont.*

Ref.	Deception-Category	Deep Learning Models Used	Key Contribution	Limitations
[31]	Moving target defense	Deep neural network, deep convolution network	They conducted a thorough study of MTD techniques, their core classifications, important design features, frequent attack behaviors addressed by current MTD techniques, and implementation found in this article.	This article only briefly investigated the relationship between MTD and other defense systems. There has been little research that looks into the influence of MTD on minimizing attacks after the reconnaissance stage. There has not been much research into the best way to use numerous hybrid MTD approaches. Existing MTD methodologies have limitations in monitoring several parameters of a system's quality.

3. Defensive Deception

3.1. Datasets Used in Defensive Deception

Table 4 below lists the various defense deception datasets utilized in various research works. Figure 4 portrays the various methods to implement Defensive Deception.

Table 4. List of various Defense Deception datasets.

Ref.	Year	Authors	Dataset Used	Dataset Size	Format	Details about the Dataset/Brief Description
[115]	2012	Ali Shiravi, Mahbod Tavallaee, Hadi Shiravi, Ali A. Ghorbani,	ISCXIDS2012	16.1 GB	Testbeds from Wireshark	This dataset was developed using a dynamic approach. Their strategy is divided into an Alpha profile and a Beta profile. The Alpha profile uses several multi-stage attack patterns to monitor the anomalous part of the dataset. On the other hand, the Beta traffic generator simulates genuine network traffic, including background noise.
[116]	2013	Gideon Creech, J. Hu	ADFA IDS	5951 records	Training and Validation type	The dataset consists of the password brute force of FTP and SSH. It also includes C100 Webshel payload, Linux Meter-preter, Java-based Meterpreter, and attack vectors with 10 attacks per vector.
[117]	1999	Salvatore J. Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis, and Philip K. Chan	KDD CUP 1999	2 million connection records with 41 features	relational	It is commonly used as a standard dataset for IDS simulations by researchers.
[118]	2000	Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, Ali A. Ghorbani	DARPA	5000 records	relational	The 1999 DARPA Intrusion Detection Examination consisted of an off-line and a real-time intrusion detection evaluation.
[119]	2016	Prudhvi Ratna Badri Satya, Kyumin Lee, Dongwon Lee, Thanh Tran, Jason (Jiasheng) Zhang	Likes of Facebook	Records including like are 13,147	relational	A study of fake Facebook Likers obtained from company employees that use the link and honeypot approaches was done. False Likers differed from genuine Likers in terms of liking behaviors, duration, etc.

3.2. Perturbation

Perturbation is a technique for limiting the leakage of sensitive data by inserting noise [20]. A defender can use perturbations to initiate Defensive Deception via external noises [19]. The method used in this article allows clients to access critical data summary facts that are not altered and do not compromise data security. Perturbation could be used to build a detailed counterplan that balances disruption with the potential to deceive the attacker based on the believability of the ploys deployed [36]. Table 5 discusses the classification of several deception categories.

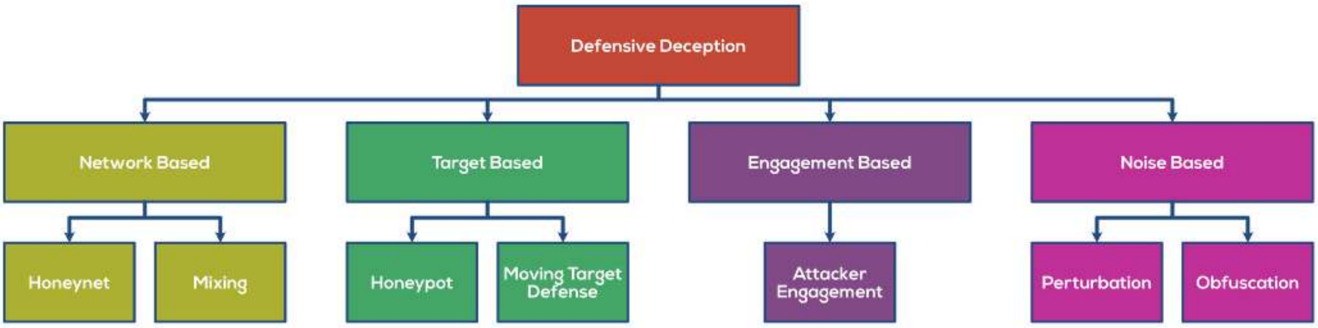

Figure 4. Methods to implement Defensive Deception.

Table 5. Classification of several deception categories.

Reference	Year	Deception Technique	Level of Interaction	Scalability	Resource Level	Goal	Main Attack	Strategy	Domain
[109]	2019	False patch technique	High	Yes	Virtual	Property preservation	Advanced persistent threats	Incorrect facts; Fraud; imitating	Game theory
[110]	2015	Honeypot, designed lure	High	Limited	Virtual	Security for assets; identification of attacks	Probing	Deceiving and imitating	Game theory
[111]	2019	Honeypot	Medium	Yes	Hybrid	Safeguarding assets; identification of attacks	DoS assaults, network drops, and APTs	Deceiving; tries to imitate	IoT
[112]	2018	Honey webs	Low	Yes	Virtual	Preservation of Assets	cyberattack	Deceiving; imitating	Cloud services from the internet
[113]	2018	Deceiving signals	Competitive	NA	Physical	Protection of resources; monitoring of attacks	Advanced persistent threats	Misguiding; concealing; imitating; deceiving	No domain name was provided.
[114]	2021	Misleading Network traffic	Dynamic/high	NA	Physical	Assets preservation	Recon/Investigating	Disguising; mirroring	Cyber–physical system
[42]	2016	Social Honeypot	High	NA	Virtual	Identifying the adversary	The malevolent demeanor of a user	Imitating	A domain is not specified

3.3. *Moving Target Defense*

Moving target defense can also be used to build an RL-CRM (Reinforcement learning—Cyber-resilient mechanisms) that attracts jammers to attack a bogus route to safeguard actual communication [15]. MTD is related to Defensive Deception in that it aims to enhance attackers' confusion or ambiguity, preventing them from escalating or failing their attacks to the next level. The major difference is that MTD does not actively mislead attackers with misleading information, whereas Defensive Deception frequently entails using fake items or details to cause aggressors to generate false ideas and be tricked into making inefficient or weak attack judgments. MTD's major trait is that it focuses on modifying system configurations with greater understanding and efficiency, whereas Defensive Deception focuses on changing the attacker's perspective [25]. An MTD's purpose in altering the Prevention Surface is to keep the attacker unsure of the defense mechanism in place, forcing the adversary to invest more resources and devise more complicated techniques to unencrypt the data. SDN (Software Defined Networking) architecture may be vertically divided into three tiers: data, control and application plane. When utilizing a Moving Target Defense that switches among several configurations, one would like to think that it improves the integrity of the implemented system while having no detrimental influence on legitimate users' efficiency. Quantitative analysis based on usability and security metrics has been used for the same. During the creation, installation, and assessment of Moving Target Defenses, their classification had aided in finding several areas that had been underexplored (MTDs). Although difficult to implement, the mobility of multiple platforms within a single framework can provide more security benefits than a single platform movement. In order to research in this field, one must first discover sets of setups that are consistent (in terms of performance) across multiple surfaces. They attempted to classify a variety of existing works using this nomenclature. E.g., in a project, a MTD is a defense used for the mobility of detecting surfaces with fixed cycle flipping framed as a multi-stage game that simulates basic use cases and assesses the security and performance of various defenses in these situations. They also discovered that hierarchical techniques such as Software Defined Networking aid in implementing MTD remedies with little networking performance effect [30]. Moving target defense can also be used to build an RL-CRM (Reinforcement learning—Cyber-resilient mechanisms) that attracts jammers to attack a bogus route to safeguard actual communication [15].

3.4. Obfuscation

Obfuscation defenses divert an enemy's resources by displaying and diverting them to decoy targets rather than the network's genuine resources and providing fake data mixed with real (i.e., valuable) data [20]. The main goal of obfuscation is to slow down the attacker's movement within the network and systems [21]. A leader–follower game (also known as the Stackelberg game) was modeled between an obfuscation technique designer and a possible attacker. To counter optimum inference attacks, the authors devised adaptive techniques. They anticipated that when consumers share sensitive data with untrustworthy entities, they will take precautions to secure it. This allows users to disguise data before sharing it by adding noises. The attacker has access to sensitive user information and obfuscation-related noises [59]. Data obfuscation has several advantages over other DD approaches, including honey-x techniques, which are designed to deceive enemies into making suboptimal or weak attack decisions by providing incorrect information, ease of deployment, and minimal cost. Adding noise to normal data, on the other hand, can confuse a defender or a legal user.

On the other hand, most data obfuscation research focuses on developing a strategy for hiding real information rather than detecting an attacker [25]. SA (Sensitivity analysis) examines how perturbed instances of the method's input affect the outcome for any particular methodology. With its random input weights, ELM produces a consistent SA, demonstrating the validity of ELM as a classifier in general and SA in particular [64].

3.5. Mixing

Mixing is a concept used in security and privacy techniques to limit likability. Mixing solutions employ exchange systems to avoid direct connectivity among networks [20]. While bait-based fraud can help increase intrusion detection by capturing additional data during the Deception, there is no guarantee of success because the assailants will not be engaged in the bait. Furthermore, if more vital information is utilized as bait to lure attackers more efficiently, the bait itself raises danger when competent attackers might deduce signs of system weaknesses based on the baits they have investigated. As a result, combining real and false information to avoid a major danger, such as semi-bait-based deceit, may be a realistic option [25,60].

3.6. Honey-x

Most honey-x tactics (e.g., honey files, honeypots, and honey tokens) are designed to deceive enemies into making suboptimal or weak attack decisions by providing incorrect information. This will necessitate the implementation of additional processes or procedures to ensure that ordinary users or defenders are not misled [25]. Honey-x deception methods related to the employment of various technologies such as honey patches, honeypots and other network assets with advanced monitoring capabilities allow network administrators to decipher details about intruders while masquerading as genuine network assets [20]. Honeypots are legal traps placed in a network to detect or deflect unauthorized access to a system. Honeypots are useful tools in understanding an attacker's intentions [22].

Honeypots are all tools that pull an attacker into a location where the security team wants them to go to assess their purpose and guide them to do things that might expose them to what they are trying to do [21].

3.7. Attacker Engagement

Attacker Engagement entails using feedback to change attacker behavior over time, squandering their efforts while enabling network managers to perform counterintelligence actions [20]. The majority of game-theoretic deception models are static games or single-shot dynamic games. However, some preliminary research has looked into multi-period games. They referred to games with many periods as "dynamic" and referred to these interactions as "attacker engagement" [6]. They used a one-sided randomized game to model the attacker. States correspond to network layers in order from left to right. As a result, the attacker remains unnoticed. Rather than ejecting the attacker, the Defense determines when to engage the attacker and gain information [60]. The authors of the paper [6] provided a list of articles that looked into mimesis. Articles mentioned by them on the left-hand side look at honey-x, whereas articles mentioned by them on the right-hand side focus on attacker engagement. There is no one-to-one correspondence between deception species and games. Two alternative methodologies are used to model honey-x. One method employs signaling games to stress the attacker's beliefs about whether systems are normal or honeypots. Bayesian Nash games are used in the other strategy. This method is based on resource allocation difficulties, and it results in an overall network design that is best for the Defense. The paper [6] lists three approaches for attacker engagement: multiple-period games, the interaction between games and MDPs (Markov Decision Process), one-sided stochastic games.

4. Open Problems in CI-Enabled Defensive Deception

The following Figure 5 illustrates the Open Problems in CI-enabled Defensive Deception.

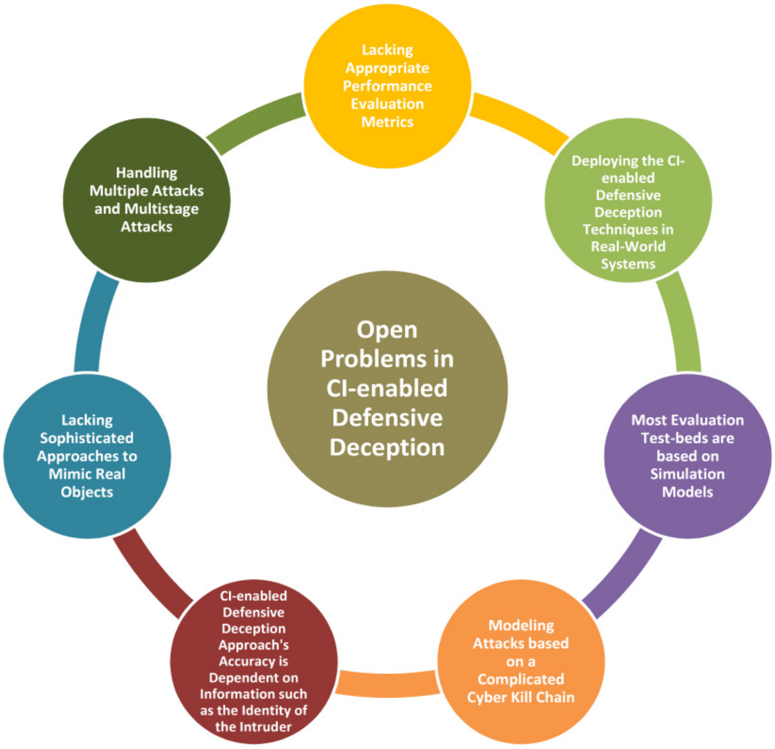
Open Problems in CI-enabled Defensive Deception
Lacking Appropriate Performance Evaluation Metrics
Deploying the CI-enabled Defensive Deception Techniques in Real-World Systems
Most Evaluation Test-beds are based on Simulation Models
Modeling Attacks based on a Complicated Cyber Kill Chain
CI-enabled Defensive Deception Approach's Accuracy is Dependent on Information such as the Identity of the Intruder
Lacking Sophisticated Approaches to Mimic Real Objects
Handling Multiple Attacks and Multistage Attacks

Figure 5. Open Problems in CI-enabled Defensive Deception.

While we can profit from modeling simple attack processes such as active reconnaissance or security breaches, deploying game-theoretic DD in real-world systems and modeling attacks based on a complicated cyber death chain remains difficult. Research has been sparse and has primarily concentrated on reconnaissance assaults in other fields. Large volumes of traffic flow data can be generated in IoT and SDN systems, which can be leveraged to train machine learning models to identify attackers. However, existing DD techniques for those domains do not include machine learning [25].

New security vulnerabilities to machine learning and deep learning algorithms emerge regularly. Even though many learning frameworks, algorithms, and optimisation mechanisms have been suggested, research into learning models' security is still in its early stages. As a result, machine learning techniques are vulnerable to various threats; hence a Defensive Deception employed with ML/DL can compromise [29].

Despite attempts to include adversarial samples in training models and improve the resilience of learning algorithms, these solutions are still incapable of solving the frequency of operation. As a result, research on safe deep learning models, such as Bayes, deep networks incorporating prior information, will be particularly intriguing soon [29].

Designing safe learning algorithms necessitates balancing security, generalization performance and cost. In general, a higher level of security results in a higher overhead or even a lower prediction accuracy of learning algorithms, which makes their implementation more difficult. Implementation of security strategies with less overhead and cost remains a challenge [29].

The efficiency of machine-learning-based deception strategies is dependent on the availability of information about the attacker, their techniques and their targets. In practice, the Defense lacks access to such information, substantially limiting the training of Machine Learning classifiers and detectors. Furthermore, models such as these are frequently presumed when the attacker behaves properly toward its intended victim. However, the efficiency of deception tactics may not be easily quantified if an attacker chooses to fool a defender to remain stealthy [75,76].

It is easier to set up simple deception tactics than configuring and applying security controls to all information systems' resources. Deception of vital information systems should be given priority. The methods are not difficult, but they require consideration when choosing from the many available options. Implementing deception on certain resources can be challenging and require much expertise to implement and maintain correctly.

The determination of time period, which is the amount of time after which we enforce the MTD in case of an attack, is in general highly specific and constant. Determining an appropriate time period that could be changed according to the situations remains a difficult task [30].

Conflicting security policies which could occur during the deployment of Defensive Deception techniques must be carefully analyzed, as such conflicts may lead to the loss of genuine user packets or the introduction of new attack vectors. Although some research has attempted to discover security policy conflict in the case of a SDN-managed cloud network, it is not immediately evident how it may be applied to MTDs [30] and other Defensive Deception technologies.

5. Future Directions in Defensive Deception

The following Figure 6 illustrates the Future Directions in CI-enabled Defensive Deception.

5.1. Honeypot

We need more parameters to judge the efficiency of honeypot techniques. Honeypot quality has been assessed mostly based on detection rate, even though Honeypot's primary function is to safeguard assets and identify threats. As a result, better metrics for measuring the responsibilities of both defending assets and detecting threats should be developed. An attacker's perceived level of uncertainty, the number of missed sensitive resources by intruders, the number of attack pathways discovered using honeypots, or the number of vital network elements assaulted are all examples of metrics. Furthermore, for the Defense to choose an appropriate method based on various variables, such as those of numerous parameters such as targets, the efficiency of running honeypot technologies, or productivity deterioration due to deception installation, must be addressed. Additionally, the attack data should be utilized to analyze intruders and generate honey sources. The only metric that captures ML-based honeypots so far is classification accuracy. Additional metrics for ML-based DD approaches, such as creating misleading traffics and network topologies, should be created. Completely automated Deception is ideal for ML-based Defensive Deception tactics such as producing deceptive traffics and network architectures.

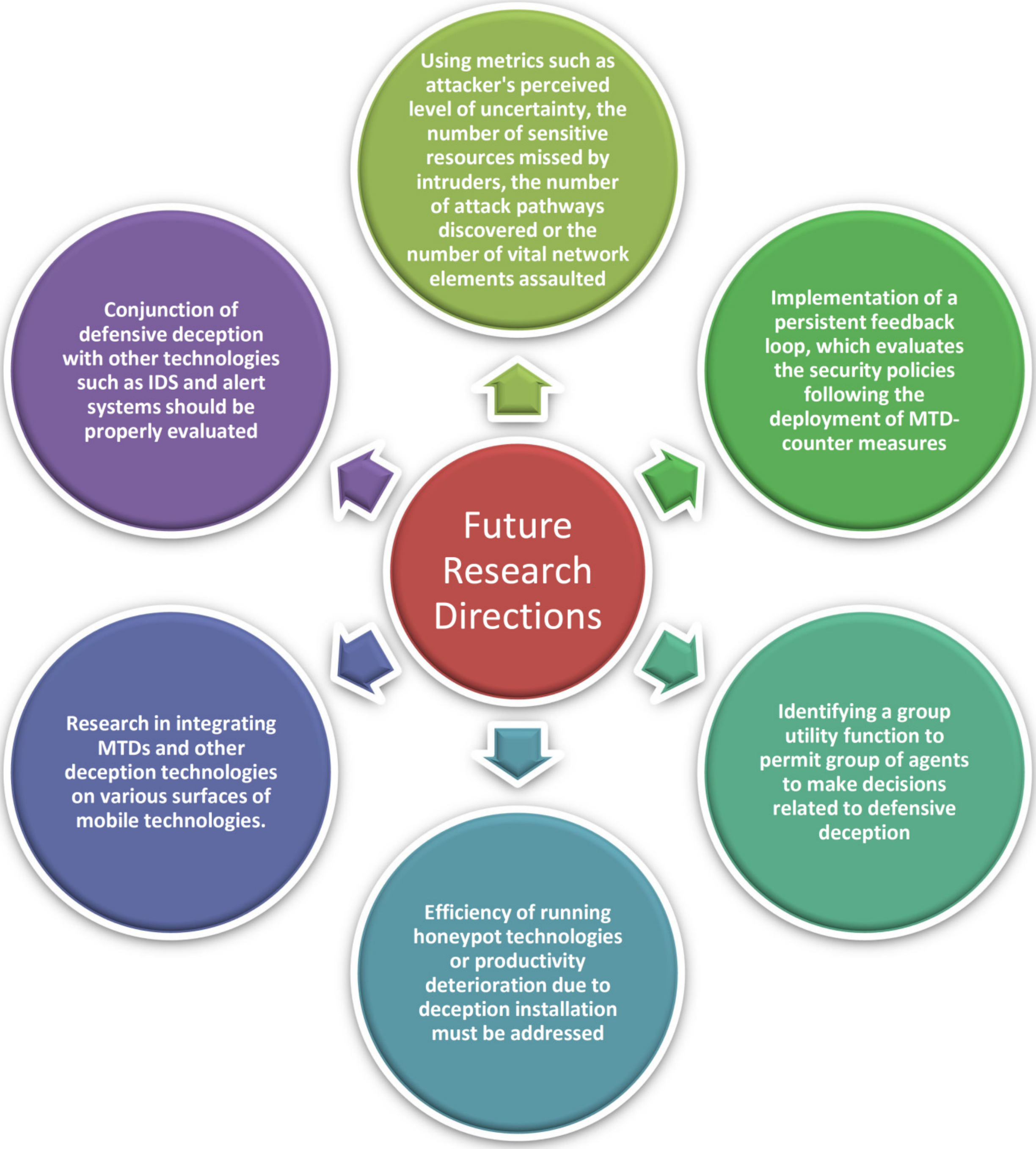
Using metrics such as attacker's perceived level of uncertainty, the number of sensitive resources missed by intruders, the number of attack pathways discovered or the number of vital network elements assaulted
Conjunction of defensive deception with other technologies such as IDS and alert systems should be properly evaluated
Implementation of a persistent feedback loop, which evaluates the security policies following the deployment of MTD-counter measures
Future Research Directions
Research in integrating MTDs and other deception technologies on various surfaces of mobile technologies.
Identifying a group utility function to permit group of agents to make decisions related to defensive deception
Efficiency of running honeypot technologies or productivity deterioration due to deception installation must be addressed

Figure 6. Future directions in Defensive Deception.

5.2. Moving Target Defense

Some of the unexplored areas of MTD that can be further researched are as follows: Most of the MTDs developed today primarily focus on computers and computer networks.

Research in integrating MTDs on various surfaces of mobile technologies and networks could be beneficial [30]. Apart from this, we need research to determine reasonable periods. Instead of keeping it constant, the time period must be changed based on the attack model [30].

5.3. Other Future Directions

The success of a Defensive Deception strategy should be measured by how successfully it misleads intruders. To judge the degree of deceit, the intruder's perspective and tactics should depend on its belief in the defender's actions. On the other hand, current research frequently uses system metrics as a substitute for evaluating the performance of a defender's deceptive strategy [72,73].

Furthermore, deception tactics should not always be used in conjunction with traditional protection services such as intrusion detection, prevention, and alert systems [74]. This is because particular combinations of deploying tactics with traditional security services, for example, employing honeypots in conjunction with intrusion detection and prevention systems, might generate an inefficient overlapping effect. As a result, we should devise a more systematic technique for using both defense services synergistically to provide cost-effective defense services [74].

When dealing with Defensive Deception technologies, a clear understanding and proper compliance of various legal security policies are difficult. To address this, we can implement a persistent feedback loop that evaluates the security policies following the deployment of MTD countermeasures. This can be accomplished by assuring end-to-end regression and integration testing for numerous network traffic instances. Another option is to simulate policy conflicts that may develop as part of the MTD modeling process. This would help us foresee the policy contradictions in case of an MTD deployment, and we can make changes accordingly [30].

Furthermore, greater research into the implications of adaptable cyber Deception and attacker expertise is required. In the future, there is a need to identify a theory of group utility function to permit groups of agents to make decisions. It will focus on automatic deception packet creation, delivery mechanism development and the resultant deception model. It will also examine if any unique defensive deception tactics or counter-deception techniques exist in the cyber environment.

6. Conclusions

The struggle for supremacy is being conducted on both sides as artificial intelligence and machine learning continue to progress at a breakneck pace for good and bad purposes [78]. With so much research and development in these fields, increased computational power, the volume and access to enormous amounts of data, and hyper interconnectivity are the mechanisms via which AI and ML advances benefit. Deception-based defenses are potent weapons that have been proven to work in various domains. Their efficacy is based on the fact that they are programmed to exploit key biases to appear realistic but misleading substitutes to the hidden reality [79–108]. As a result, one will require a thorough understanding of both offensive and defensive trickery to implement a perfect Deception strategy.

Such methods give defenders a tactical advantage by learning further about their enemies, limiting secondary information breaches in their systems, and better understanding their attackers. The effects of adaptive Defensive Deception on an automated attacker are compared in this study. This article demonstrated how an autonomic system could manage a Defensive Deception system. As examples, certain procedures and methods are offered to investigate proposals and solutions to specific fraud detection and prevention issues.

AI/MLS models have already proven to be a benefit and a burden in the cybersecurity field. Consequently, current cybersecurity measures are expected to become obsolete, forcing the creation of new countermeasures. Deception tactics based on machine learning that learn and recognize can be a great tool to automatically deploy and maintain the

Deception frameworks. Machine Learning can improve Deception by taking into account various factors such as:

1. We must think about the types of datasets used to construct deception tactics.
2. Good datasets for replicating actual things and evaluating false items are required to create plausible fake objects.
3. ML-based deception tactics should use appropriate metrics to capture their effectiveness and efficiency.

Unlike typical defense methods, Deception entails some risk because it necessitates certain contacts with attackers to confuse or mislead them. It is unavoidable to accept the risk if the purpose of protection necessitates long-term Deception. As mentioned in our future direction, DD should be used with other legacy defensive techniques such as intrusion prevention or detection with proper precations to minimize an overabundance of risk. Moving target defense (MTD) or obfuscation tactics have a similar purpose to Deception in creating attackers' confusion or doubt. Deception, on the other hand, would produce fake objects or information to deceive an attacker's cognitive perspective or create a false notion, causing the attacker to pick a sub-optimal or bad attack technique, except for obfuscation or MTD, which modifies configuration settings or data based on the current resources of a system. Hence these properties must be considered when we want to deploy MTD or obfuscation as defence [72,77]. Overall, when we employ CI-enabled techniques such as ML/DL in proper conjunction with Defensive Deception, we can safely protect our resources in a very effective manner. However, at the same time, when the CI techniques are implemented blindly without proper consideration of resources or a final goal, it would not result in decreased protection but would also cause a huge wastage of resources.

Author Contributions: Conceptualization, K.S.; methodology, K.S.; software, U.C.; validation, J.T.S.; formal analysis, A.G.; investigation, P.V.M. and S.D.; resources, U.C., and K.S.; data curation, A.G.; writing—original draft preparation, P.V.M., S.D., A.G. and K.S.; writing—review and editing, U.C., K.S. and J.T.S.; visualization, U.C. and K.S.; supervision, K.S.; project administration, J.T.S.; funding acquisition, J.T.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1A2C1012187, 50%), the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021000493, 5G Massive Next Generation Cyber Attack Deception Technology Development, 50%).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Li, J.H. Cyber security meets artificial intelligence: A survey. *Front. Inf. Technol. Electron. Eng.* **2018**, *19*, 1462–1474. [CrossRef]
2. Yang, K.C.; Varol, O.; Davis, C.A.; Ferrara, E.; Flammini, A.; Menczer, F. Arming the public with artificial intelligence to counter social bots. *Hum. Behav. Emerg. Technol.* **2019**, *1*, 48–61. [CrossRef]
3. Jean-Philippe, R. Enhancing Computer Network Defense Technologies with Machine Learning and Artificial Intelligence. Ph.D. Thesis, Utica College, Utica, NY, USA, 2018.
4. Dilek, S.; Çakır, H.; Aydın, M. Applications of artificial intelligence techniques to combating cyber crimes: A review. *arXiv* **2015**, arXiv:1502.03552. [CrossRef]
5. Zeadally, S.; Adi, E.; Baig, Z.; Khan, I.A. Harnessing artificial intelligence capabilities to improve cybersecurity. *IEEE Access* **2020**, *8*, 23817–23837. [CrossRef]
6. Pawlick, J.; Colbert, E.; Zhu, Q. A game-theoretic taxonomy and survey of Defensive Deception for cybersecurity and privacy. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–28. [CrossRef]
7. Layton, P. Fighting Artificial Intelligence Battles: Operational Concepts for Future AI-Enabled Wars. *Network* **2021**, *4*, 20.

8. Landsborough, J.; Carpenter, L.; Coronado, B.; Fugate, S.; Ferguson-Walter, K.; Van Bruggen, D. Towards Self-Adaptive Cyber Deception for Defense. In Proceedings of the HICSS (Hawaii International Conference on System Sciences), Online, 5–8 January 2021; pp. 1–10.
9. King, T.C.; Aggarwal, N.; Taddeo, M.; Floridi, L. Artificial intelligence crime: An interdisciplinary analysis of foreseeable threats and solutions. *Sci. Eng. Ethics* **2020**, *26*, 89–120. [CrossRef]
10. Kotenko, I. Multi-agent modelling and simulation of cyber-attacks and cyber-defense for homeland security. In Proceedings of the 2007 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Dortmund, Germany, 6–8 September 2007; pp. 614–619.
11. Rowe, N.C.; Rothstein, H.S. Two taxonomies of Deception for attacks on information systems. *J. Inf. Warf.* **2004**, *3*, 27–39.
12. Behdad, M.; Barone, L.; Bennamoun, M.; French, T. Nature-inspired techniques in the context of fraud detection. *IEEE Trans. Syst. Man Cybern. Part C* **2012**, *42*, 1273–1290. [CrossRef]
13. Happa, J.; Bashford-Rogers, T.; van Rensburg, A.J.; Goldsmith, M.; Creese, S. Deception in Network Defenses using unpredictability. *Digit. Threat. Res. Pract.* **2021**, *2*, 29. [CrossRef]
14. Vinayakumar, R.; Soman, K.P.; Poornachandran, P.; Sachin Kumar, S. Detecting Android malware using long short-term memory (LSTM). *J. Intell. Fuzzy Syst.* **2018**, *34*, 1277–1288. [CrossRef]
15. Huang, Y.; Huang, L.; Zhu, Q. Reinforcement learning for feedback-enabled cyber resilience. *arXiv* **2021**, arXiv:2107.00783. [CrossRef]
16. Sadgali, I.; Sael, N.; Benabbou, F. Performance of machine learning techniques in the detection of financial frauds. *Procedia Comput. Sci.* **2019**, *148*, 45–54. [CrossRef]
17. Xiao, Q. Technology review-biometrics-technology, application, challenge, and computational intelligence solutions. *IEEE Comput. Intell. Mag.* **2007**, *2*, 5–25. [CrossRef]
18. Hassan, A.K.I.; Abraham, A. Computational intelligence models for insurance fraud detection: A review of a decade of research. *J. Netw. Innov. Comput.* **2013**, *1*, 341–347.
19. Huang, L.; Zhu, Q. A dynamic games approach to proactive defense strategies against advanced persistent threats in cyber-physical systems. *Comput. Secur.* **2020**, *89*, 101660. [CrossRef]
20. Cifranic, N.; Hallman, R.A.; Romero-Mariona, J.; Souza, B.; Calton, T.; Coca, G. Decepti-SCADA: A cyber deception framework for active Defense of networked critical infrastructures. *Internet Things* **2020**, *12*, 100320. [CrossRef]
21. Gurr, J.J. Deceptive Machine Learning for Offense and Defense Targeting Financial Institutions. Ph.D. Thesis, Utica College, Utica, NY, USA, 2018.
22. Kiwia, D.; Dehghantanha, A.; Choo, K.K.R.; Slaughter, J. A cyber kill chain based taxonomy of banking Trojans for evolutionary computational intelligence. *J. Comput. Sci.* **2018**, *27*, 394–409. [CrossRef]
23. Rowe, N.C. A model of Deception during cyber-attacks on information systems. In Proceedings of the IEEE First Symposium onMulti-Agent Security and Survivability, Drexel, PA, USA, 31 August 2004; pp. 21–30.
24. Sharma, A.; Panigrahi, P.K. A review of financial accounting fraud detection based on data mining techniques. *arXiv* **2013**, arXiv:1309.3944. [CrossRef]
25. Zhu, M.; Anwar, A.H.; Wan, Z.; Cho, J.H.; Kamhoua, C.A.; Singh, M.P. A survey of Defensive Deception: Approaches using game theory and machine learning. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 2460–2493. [CrossRef]
26. Kamoun, F.; Iqbal, F.; Esseghir, M.A.; Baker, T. AI and machine learning: A mixed blessing for cybersecurity. In Proceedings of the 2020 International Symposium on Networks, Computers and Communications (ISNCC), Montreal, QC, Canada, 20–22 October 2020; pp. 1–7.
27. Demertzis, K.; Iliadis, L. A bio-inspired hybrid artificial intelligence framework for cyber security. In *Computation, Cryptography, and Network Security*; Springer: Cham, Switzerland, 2015; pp. 161–193.
28. Goethals, P.L.; Hunt, M.E. A review of scientific research in defensive cyberspace operation tools and technologies. *J. Cyber Secur. Technol.* **2019**, *3*, 1–46. [CrossRef]
29. Liu, Q.; Li, P.; Zhao, W.; Cai, W.; Yu, S.; Leung, V.C. A survey on security threats and defensive techniques of machine learning: A data driven view. *IEEE Access* **2018**, *6*, 12103–12117. [CrossRef]
30. Sengupta, S.; Chowdhary, A.; Sabur, A.; Alshamrani, A.; Huang, D.; Kambhampati, S. A survey of moving target defenses for network security. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1909–1941. [CrossRef]
31. Cho, J.H.; Sharma, D.P.; Alavizadeh, H.; Yoon, S.; Ben-Asher, N.; Moore, T.J.; Nelson, F.F. Toward proactive, adaptive Defense: A survey on moving target defense. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 709–745. [CrossRef]
32. Sharma, P.; Sarma, K.K.; Mastorakis, N.E. Artificial Intelligence Aided Electronic Warfare Systems-Recent Trends and Evolving Applications. *IEEE Access* **2020**, *8*, 224761–224780. [CrossRef]
33. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
34. Mohanta, B.K.; Jena, D.; Satapathy, U.; Patnaik, S. Survey on IoT security: Challenges and solution using machine learning, artificial intelligence and blockchain technology. *Internet Things* **2020**, *11*, 100227. [CrossRef]
35. Gupta, C.; Johri, I.; Srinivasan, K.; Hu, Y.-C.; Qaisar, S.M.; Huang, K.-Y. A Systematic Review on Machine Learning and Deep Learning Models for Electronic Information Security in Mobile Networks. *Sensors* **2022**, *22*, 2017. [CrossRef]

36. Michael, J.B.; Rowe, N.C.; Auguston, M.; Drusinsky, D.; Rothstein, H.S. *Phase II Report on Intelligent Software Decoys: Intelligent Software Decoy Tools for Cyber Counterintelligence and Security Countermeasures*; Department of Computer Science, Naval Postgraduate School: Monterey, CA, USA, 2004.
37. Na, S.; Xumin, L.; Yong, G. Research on k-means clustering algorithm: An improved k-means clustering algorithm. In Proceedings of the 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, Jian, China, 2–4 April 2010; pp. 63–67.
38. Alom, M.Z.; Taha, T.M. Network intrusion detection for cyber security using unsupervised deep learning approaches. In Proceedings of the 2017 IEEE National Aerospace and Electronics Conference (NAECON), Dayton, OH, USA, 27–30 June 2017; pp. 63–69.
39. Kausar, N.; Samir, B.B.; Abdullah, A.; Ahmad, I.; Hussain, M. A review of classification approaches using support vector machine in intrusion detection. In Proceedings of the International Conference on Informatics Engineering and Information Science, Kuala Lumpur, Malaysia, 12–14 November 2011; Springer: Berlin/Heidelberg, Germany; pp. 24–34.
40. Champaneria, P.; Shah, B.; Panchal, K. Survey on intrusion detection system using support vector machine. *Int. J. Emerg. Technol. Adv. Eng.* **2014**, *4*, 220–225.
41. Manekar, V.; Waghmare, K. Intrusion detection system using support vector machine (SVM) and particle swarm optimization (PSO). *Int. J. Adv. Comput. Res.* **2014**, *4*, 808.
42. Nisrine, M. A security approach for social networks based on honeypots. In Proceedings of the 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt), Tangier, Morocco, 24–26 October 2016; pp. 638–643.
43. Zhu, H. Fighting against Social Spammers on Twitter by Using Active Honeypots. Master's Thesis, McGill University, Montreal, QC, Canada, 2014.
44. Burkard, C.; Lagesse, B. Analysis of causative attacks against svms learning from data streams. In Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics, Scottsdale, AZ, USA, 24 March 2017; pp. 31–36.
45. Yang, C.; Wu, Q.; Li, H.; Chen, Y. Generative poisoning attack method against neural networks. *arXiv* **2017**, arXiv:1703.01340.
46. Kim, J.; Kim, H. Applying recurrent neural network to intrusion detection with hessian free optimization. In *International Workshop on Information Security Applications*; Springer: Cham, Switzerland, 2015; pp. 357–369.
47. Li, X.; Huang, Z.; Wang, F.; Wang, X.; Liu, T. Toward convolutional neural networks on pulse repetition interval modulation recognition. *IEEE Commun. Lett.* **2018**, *22*, 2286–2289. [CrossRef]
48. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
49. Lee, G.H.; Jo, J.; Park, C.H. Jamming prediction for radar signals using machine learning methods. *Secur. Commun. Netw.* **2020**, *2020*, 2151570. [CrossRef]
50. Kang, J.; Jang, S.; Li, S.; Jeong, Y.S.; Sung, Y. Long short-term memory-based malware classification method for information security. *Comput. Electr. Eng.* **2019**, *77*, 366–375. [CrossRef]
51. Bengio, Y.; Delalleau, O. On the expressive power of deep architectures. In Proceedings of the International Conference on Algorithmic Learning Theory, Espoo, Finland, 5–7 October 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 18–36.
52. Yi, H.; Shiyu, S.; Xiusheng, D.; Zhigang, C. A study on deep neural networks framework. In Proceedings of the 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 3–5 October 2016; pp. 1519–1522.
53. Sengupta, S.; Chakraborti, T.; Kambhampati, S. Mtdeep: Boosting the security of deep neural nets against adversarial attacks with moving target defense. In Proceedings of the Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence, Orleans, LA, USA, 2–7 February 2018.
54. Wei, W.; Liu, L. Robust deep learning ensemble against Deception. *IEEE Trans. Dependable Secur. Comput.* **2020**, *18*, 1513–1527. [CrossRef]
55. Gu, S.; Rigazio, L. Towards deep neural network architectures robust to adversarial examples. *arXiv* **2014**, arXiv:1412.5068.
56. Li, Y.; Wang, X.; Liu, D.; Guo, Q.; Liu, X.; Zhang, J.; Xu, Y. On the performance of deep reinforcement learning-based anti-jamming method confronting intelligent jammer. *Appl. Sci.* **2019**, *9*, 1361. [CrossRef]
57. Liu, Y.; Wang, H.; Peng, M.; Guan, J.; Xu, J.; Wang, Y. DeePGA: A privacy-preserving data aggregation game in crowdsensing via deep reinforcement learning. *IEEE Internet Things J.* **2019**, *7*, 4113–4127. [CrossRef]
58. Xu, Q.; Su, Z.; Lu, R. Game theory and reinforcement learning based secure edge caching in mobile social networks. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3415–3429. [CrossRef]
59. Shokri, R. Privacy games: Optimal user-centric data obfuscation. *arXiv* **2014**, arXiv:1402.3426. [CrossRef]
60. Horák, K.; Zhu, Q.; Bošanský, B. Manipulating adversary's belief: A dynamic game approach to Deception by design for proactive network security. In Proceedings of the International Conference on Decision and Game Theory for Security, Vienna, Austria, 23–25 October 2017; Springer: Cham, Switzerland, 2017; pp. 273–294.
61. Lee, T.; Edwards, B.; Molloy, I.; Su, D. Defending against neural network model stealing attacks using deceptive perturbations. In Proceedings of the 2019 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 19–23 May 2019; pp. 43–49.
62. Fraunholz, D.; Anton, S.D.; Lipps, C.; Reti, D.; Krohmer, D.; Pohl, F.; Schotten, H.D. Demystifying deception technology: A survey. *arXiv* **2018**, arXiv:1804.06196.
63. Al-Shaer, E.; Wei, J.; Kevin, W.; Wang, C. *Autonomous Cyber Deception*; Springer: Berlin/Heidelberg, Germany, 2019.

64. Sun, F.; Toh, K.A.; Romay, M.G.; Mao, K. (Eds.) *Extreme Learning Machines 2013: Algorithms and Applications*; Springer International Publishing: Berlin/Heidelberg, Germany, 2014.
65. Nanda, S.; Zafari, F.; De Cusatis, C.; Wedaa, E.; Yang, B. Predicting network attack patterns in SDN using machine learning approach. In Proceedings of the 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Palo Alto, CA, USA, 7–10 November 2016; pp. 167–172.
66. Thejas, G.S.; Soni, J.; Chandna, K.; Iyengar, S.S.; Sunitha, N.R.; Prabakar, N. Learning-Based Model to Fight against Fake Like Clicks on Instagram Posts. In Proceedings of the 2019 SoutheastCon, Huntsville, AL, USA, 11–14 April 2019.
67. Mashima, D. MITRE ATT&CK Based Evaluation on In-Network Deception Technology for Modernized Electrical Substation Systems. *Sustainability* **2022**, *14*, 1256. [CrossRef]
68. Hofer, W.; Edgar, T.; Vrabie, D.; Nowak, K. Model-driven Deception for Control System Environments. In Proceedings of the 2019 IEEE International Symposium on Technologies for Homeland Security (HST), Woburn, MA, USA, 5–6 November 2019; pp. 1–7.
69. Li, H.; Guo, Y.; Sun, P.; Wang, Y.; Huo, S. An optimal Defensive Deception framework for the container-based cloud with deep reinforcement learning. *IET Inf. Secur.* **2021**, 1–15. [CrossRef]
70. Ayoade, G.; Araujo, F.; Al-Naami, K.; Mustafa, A.M.; Gao, Y.; Hamlen, K.W.; Khan, L. Automating Cyberdeception Evaluation with Deep Learning. In Proceedings of the Hawaii International Conference on System Sciences 2020 (HICSS-53), Maui, HI, USA, 7–10 January 2020; Volume 3, pp. 1–10.
71. Dlamini, M.T.; Venter, H.S.; Eloff, J.H.; Eloff, M. Digital Deception in cybersecurity: An information behaviour lens. In Proceedings of the Information Behaviour Conference, Pretoria, South Africa, 28 September–1 October 2020.
72. Datta, D.; Garg, L.; Srinivasan, K.; Inoue, A.; Reddy, G.T.; Reddy, M.P.K.; Ramesh, K.; Nasser, N. An efficient sound and data steganography based secure authentication system. *Comput. Mater. Contin.* **2021**, *67*, 723–751. [CrossRef]
73. Patel, D.; Srinivasan, K.; Chang, C.-Y.; Gupta, T.; Kataria, A. Network Anomaly Detection inside Consumer Networks—A Hybrid Approach. *Electronics* **2020**, *9*, 923. [CrossRef]
74. Sriram, P.P.; Wang, H.C.; Jami, H.G.; Srinivasan, K. 5G Security: Concepts and Challenges. In *5G Enabled Secure Wireless Networks*; Jayakody, D., Srinivasan, K., Sharma, V., Eds.; Springer: Cham, Switzerland, 2019. [CrossRef]
75. Srinivasan, K.; Gupta, T.; Agarwal, P.; Nema, A. A robust security framework for cloud-based logistics services. In Proceedings of the 2018 IEEE International Conference on Applied System Invention (ICASI), Chiba, Japan, 13–17 April 2018; pp. 162–165. [CrossRef]
76. Choudhury, M.; Srinivasan, K. An Overview into the Aspects of Fake Product Reviews, its Manipulation, and its Effects and Monitoring. In Proceedings of the 2019 IEEE International Conference on Consumer Electronics—Taiwan (ICCE-TW), Yilan, Taiwan, 20–22 May 2019; pp. 1–2. [CrossRef]
77. Srinivasan, K.; Gowthaman, T.; Kanakaraj, J. A novel copyright marking approach using steganography and robust RSA asymmetric-key cryptographic technique in audio files. *J. Discret. Math. Sci. Cryptogr.* **2017**, *20*, 1563–1571. [CrossRef]
78. Akshay Kumaar, M.; Samiayya, D.; Vincent, P.M.D.R.; Srinivasan, K.; Chang, C.-Y.; Ganesh, H. A Hybrid Framework for Intrusion Detection in Healthcare Systems Using Deep Learning. *Front. Public Health* **2022**, *9*, 824898. [CrossRef]
79. Steingartner, W.; Galinec, D.; Kozina, A. Threat Defense: Cyber Deception Approach and Education for Resilience in Hybrid Threats Model. *Symmetry* **2021**, *13*, 597. [CrossRef]
80. Seo, S.; Kim, D. OSINT-Based LPC-MTD and HS-Decoy for Organizational Defensive Deception. *Appl. Sci.* **2021**, *11*, 3402. [CrossRef]
81. Seo, S.; Kim, D. SOD2G: A Study on a Social-Engineering Organizational Defensive Deception Game Framework through Optimization of Spatiotemporal MTD and Decoy Conflict. *Electronics* **2021**, *10*, 3012. [CrossRef]
82. Wang, C.; Zeng, C.; Liu, H.; Chen, J. Adversarial Hiding Deception Strategy and Network Optimization Method for Heterogeneous Network Defense. *Electronics* **2021**, *10*, 2614. [CrossRef]
83. Park, K.; Woo, S.; Moon, D.; Choi, H. Secure Cyber Deception Architecture and Decoy Injection to Mitigate the Insider Threat. *Symmetry* **2018**, *10*, 14. [CrossRef]
84. Gallardo-Antolín, A.; Montero, J.M. Detecting Deception from Gaze and Speech Using a Multimodal Attention LSTM-Based Framework. *Appl. Sci.* **2021**, *11*, 6393. [CrossRef]
85. Zeng, C.; Ren, B.; Liu, H.; Chen, J. Applying the Bayesian Stackelberg Active Deception Game for Securing Infrastructure Networks. *Entropy* **2019**, *21*, 909. [CrossRef]
86. Park, J.-G.; Lee, Y.; Kang, K.-W.; Lee, S.-H.; Park, K.-W. Ghost-MTD: Moving Target Defense via Protocol Mutation for Mission-Critical Cloud Systems. *Energies* **2020**, *13*, 1883. [CrossRef]
87. Jiang, P.; Huang, S.; Zhang, T. Optimal Deception Strategies in Power System Fortification against Deliberate Attacks. *Energies* **2019**, *12*, 342. [CrossRef]
88. Yang, Y.; Che, B.; Zeng, Y.; Cheng, Y.; Li, C. MAIAD: A Multi-stage Asymmetric Information Attack and Defense Model Based on Evolutionary Game Theory. *Symmetry* **2019**, *11*, 215. [CrossRef]
89. Shi, L.; Wang, X.; Hou, H. Research on Optimization of Array Honeypot Defense Strategies Based on Evolutionary Game Theory. *Mathematics* **2021**, *9*, 805. [CrossRef]
90. Al-Jaoufi, M.A.A.; Liu, Y.; Zhang, Z. An Active Defense Model with Low Power Consumption and Deviation for Wireless Sensor Networks Utilizing Evolutionary Game Theory. *Energies* **2018**, *11*, 1281. [CrossRef]

91. Wang, K.; Tong, M.; Yang, D.; Liu, Y. A Web-Based Honeypot in IPv6 to Enhance Security. *Information* **2020**, *11*, 440. [CrossRef]
92. Li, Y.; Shi, L.; Feng, H. A Game-Theoretic Analysis for Distributed Honeypots. *Future Internet* **2019**, *11*, 65. [CrossRef]
93. Diamantoulakis, P.; Dalamagkas, C.; Radoglou-Grammatikis, P.; Sarigiannidis, P.; Karagiannidis, G. Game Theoretic Honeypot Deployment in Smart Grid. *Sensors* **2020**, *20*, 4199. [CrossRef] [PubMed]
94. Ismailov, M.; Tsikerdekis, M.; Zeadally, S. Vulnerabilities to Online Social Network Identity Deception Detection Research and Recommendations for Mitigation. *Future Internet* **2020**, *12*, 148. [CrossRef]
95. Zhao, F.; Yuan, J.; Wang, N.; Zhang, Z.; Wen, H. Secure Load Frequency Control of Smart Grids under Deception Attack: A Piecewise Delay Approach. *Energies* **2019**, *12*, 2266. [CrossRef]
96. Bonguet, A.; Bellaiche, M. A Survey of Denial-of-Service and Distributed Denial of Service Attacks and Defenses in Cloud Computing. *Future Internet* **2017**, *9*, 43. [CrossRef]
97. Qiu, S.; Liu, Q.; Zhou, S.; Wu, C. Review of Artificial Intelligence Adversarial Attack and Defense Technologies. *Appl. Sci.* **2019**, *9*, 909. [CrossRef]
98. Li, Y.; Wang, Y. Defense against Adversarial Attacks in Deep Learning. *Appl. Sci.* **2019**, *9*, 76. [CrossRef]
99. Park, B.-S.; Yoo, S.-J. Adaptive Secure Control for Leader-Follower Formation of Nonholonomic Mobile Robots in the Presence of Uncertainty and Deception Attacks. *Mathematics* **2021**, *9*, 2190. [CrossRef]
100. Tang, L.; Mahmoud, Q.H. A Survey of Machine Learning-Based Solutions for Phishing Website Detection. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 672–694. [CrossRef]
101. Yang, P.; Gao, F.; Zhang, H. Multi-Player Evolutionary Game of Network Attack and Defense Based on System Dynamics. *Mathematics* **2021**, *9*, 3014. [CrossRef]
102. Truong, T.C.; Diep, Q.B.; Zelinka, I. Artificial Intelligence in the Cyber Domain: Offense and Defense. *Symmetry* **2020**, *12*, 410. [CrossRef]
103. Sadik, S.; Ahmed, M.; Sikos, L.F.; Islam, A.K.M.N. Toward a Sustainable Cybersecurity Ecosystem. *Computers* **2020**, *9*, 74. [CrossRef]
104. Merrick, K.; Hardhienata, M.; Shafi, K.; Hu, J. A Survey of Game Theoretic Approaches to Modelling Decision-Making in Information Warfare Scenarios. *Future Internet* **2016**, *8*, 34. [CrossRef]
105. Georgiadou, A.; Mouzakitis, S.; Askounis, D. Assessing MITRE ATT&CK Risk Using a Cyber-Security Culture Framework. *Sensors* **2021**, *21*, 3267. [CrossRef] [PubMed]
106. Demertzis, K.; Tziritas, N.; Kikiras, P.; Sanchez, S.L.; Iliadis, L. The Next Generation Cognitive Security Operations Center: Adaptive Analytic Lambda Architecture for Efficient Defense against Adversarial Attacks. *Big Data Cogn. Comput.* **2019**, *3*, 6. [CrossRef]
107. Joung, J.; Choi, J.; Jung, B.C.; Yu, S. Artificial noise injection and its power loading methods for secure space-time line coded systems. *Entropy* **2019**, *21*, 515. [CrossRef]
108. Tseng, S.M.; Chen, Y.F.; Tsai, C.S.; Tsai, W.D. Deep-learning-aided cross-layer resource allocation of OFDMA/NOMA video communication systems. *IEEE Access* **2019**, *7*, 157730–157740. [CrossRef]
109. Cho, J.-H.; Zhu, M.; Singh, M.P. *Modeling and Analysis of Deception Games Based on Hypergame Theory*; Springer Nature: Cham, Switzerland, 2019; Chapter 4; pp. 49–74. [CrossRef]
110. Kiekintveld, C.; Lisý, V.; Píbil, R. Game-theoretic foundations for the strategic use of honeypots in network security. In *Cyber Warfare*; Springer: Cham, Switzerland, 2015; pp. 81–101. [CrossRef]
111. Sengupta, S.; Chowdhary, A.; Huang, D.; Kambhampati, S. General sum markov games for strategic detection of advanced persistent threats using moving target defense in cloud networks. In Proceedings of the International Conference on Decision and Game Theory for Security, Stockholm, Sweden, 30 October–1 November 2019; Springer: Cham, Switzerland, 2019; pp. 492–512. [CrossRef]
112. El-Kosairy, A.; Azer, M.A. A new Web deception system framework. In Proceedings of the 2018 1st International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 4–6 April 2018; pp. 1–10. [CrossRef]
113. Huang, L.; Zhu, Q. Dynamic bayesian games for adversarial and defensive cyber deception. In *Autonomous Cyber Deception*; Springer: Cham, Switzerland, 2019; pp. 75–97. [CrossRef]
114. Sayin, M.O.; Başar, T. Deception-as-defense framework for cyber-physical systems. In *Safety, Security and Privacy for Cyber-Physical Systems*; Springer: Cham, Switzerland, 2021; pp. 287–317. [CrossRef]
115. Available online: https://www.unb.ca/cic/datasets/ids.html (accessed on 12 December 2021).
116. Creech, G.; Hu, J. Generation of a new IDS test dataset: Time to retire the KDD collection. In Proceedings of the 2013 IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, China, 7–10 April 2013; pp. 4487–4492. [CrossRef]
117. KDD Cup. University of California, Irvine (UCI). 1999. Available online: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed on 12 December 2021).
118. Available online: https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset (accessed on 12 December 2021).
119. Available online: http://digital.cs.usu.edu/%CB%9Ckyumin/data.html (accessed on 12 December 2021).
120. Shrivastava, R.K.; Ramakrishna, S.; Hota, C. Game Theory based Modified Naïve-bayes Algorithm to detect DoS attacks using Honeypot. In Proceedings of the 2019 IEEE 16th India Council International Conference (INDICON), Rajkot, India, 13–15 December 2019; pp. 1–4.

121. Rowe, N.C. Honeypot deception tactics. In *Autonomous Cyber Deception*; Springer: Cham, Switzerland, 2019; pp. 35–45.

122. Srivastava, N.; Dubey, S. Deception detection using artificial neural network and support vector machine. In Proceedings of the 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 29–31 March 2018; pp. 1205–1208.

123. Oluoha, O.U.; Yange, T.S.; Okereke, G.E.; Bakpo, F.S. Cutting Edge Trends in Deception Based Intrusion Detection Systems—A Survey. *J. Inf. Secur.* **2021**, *12*, 250–269. [CrossRef]