International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5793
***Article Received**: 07 August 2022 **Revised**: 10 October 2022 **Accepted**: 06 November 2022*

A Novel Ensemble Model Using Learning Classifiers to Enhance Malware Detection for Cyber Security Systems

Abhijit Das[1], Pramod[2]

[1]Research Scholar, VTU, Department of CSE, PES Institute of Technology & Management, Affiliated to VTU, Shimoga, Karnataka, INDIA
[2]Associate Professor, Department of ISE, PES Institute of Technology & Management, Affiliated to VTU, Shimoga, Karnataka, INDIA
E-mail: [1]abhijit.tec@gmail.com , [2]pramod741230@gmail.com

Abstract
In the Internet of Things arena, smart gadgets are employed to offer quick and dependable access to services. IoT technology has the ability to recognize extensive information, provide information reliably, and process that information intelligently. Data networks, controllers, and sensors are increasingly used in industrial systems nowadays. Attacks have increased as a result of the growth in connected systems and the technologies they employ. These attacks may interrupt international business and result in significant financial losses. Utilizing a variety of methods, including deep learning (DL) and machine learning (ML), cyber assaults have been discovered. In this research, we provide an ensemble staking approach to efficiently and quickly detect cyber-attacks in the IoT. The NSL, credit card, and UNSW information bases were the three separate datasets used for the experiments. The suggested novel combinations of ensemble classifiers are done better than the other individual classifiers from the base model. Additionally, based on the test outcomes, it could be concluded that all tree and bagging-based combinations performed admirably and that, especially when their corresponding hyperparameters are set properly, differences in performance across methods are not significant statistically. Additionally, compared to other comparable PE (Portable Executable) malware detectors that were published recently, the suggested tree-based ensemble approaches outperformed them.

Keywords: *Cybersecurity, malware detection, IoT, ensemble, portable executable-PE malware, hyper-parameters, and bagging*

1. INTRODUCTION

Various sorts of security events, such as unapproved users [1], data leaks [5], malware attacks [3], social engineering or hacking [6], zero-day assaults [4], denial of service (DoS) [2], etc., have expanded at an incredible rate over the last century. Due to the rising of information systems. Less than 50 million malware applications were recorded in 2010.

This estimated population increased to over 100 million in 2012. According to AV-TEST data, over 900 million malware applications were found to be in circulation in 2019, and this figure is steadily growing [7]. Network and cybercrime assaults may lead to considerable lost profits for both individuals and corporations. Data breaches are expensive, costing USD 8.191 million globally [8] and USD 3.8 million in the US, while cybercrime loses economic growth of USD 40.0 million. Cybersecurity budgets the global economy USD 401 billion yearly [9], even though an information leak costs USD 42 million in the United States and USD 8.29 million elsewhere [8]. In five years, the number of destroyed statistics is expected to triple [10]. As a consequence, companies should implement and create a robust cybersecurity strategy to prevent such losses. According to a recent sociological study, the nation's future is contingent on the government's public with information access programs and high-security equipment [11], 0 billion per year [9]. The intelligence community anticipates that the number of records broken will triple over the following 5 years [10]. As an outcome, businesses should implement and create an effective security strategy to prevent such losses. The nation's upcoming is dependent on the government, persons with the right to use high-security equipment, and knowledge programmes, according to a recent sociological study [11]. Additionally, it depends on companies granting access to their staff members, who are capable of and knowledgeable about recognizing such cyber threats immediately and efficiently. Therefore, the main issue that requires urgent attention is the need to proactively recognize different cyber events, whether known or unknown, and appropriately protect key systems from such cyber-attacks.

Cybersecurity is the field of technology and practices that guard against unwanted access, attacks, and damage to systems, networking, applications, and information [12]. Cybersecurity includes corporate to mobile computers and may be segmented. These include (i) network infrastructure, where it aims to avoid hacking or intrusions from accessing a computer server; (ii) vulnerability scanning, which considers software safe and protection tools from dangers or cyber-

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5793
***Article Received**: 07 August 2022 **Revised**: 10 October 2022 **Accepted**: 06 November 2022*

threats; (iii) knowledge confidence, where it mainly considers privacy and safeguards of valuable information; and (iv) safekeeping procedures, which makes reference to the guidelines for controlling and conserving data resources. A firewall, antiviral application, or intrusion detection software are common networking and computer monitoring security procedures. The movement is being spearheaded by digital marketing, where machine learning, a critical element of "Artificial Intelligence," may be essential in finding hidden patterns in information. Data analysis is creating a new perspective, and deep learning has influenced cybercrime [13,14]. As noted in [15], the improvement of cyber-attack techniques has made hackers more successful, resulting in an increase in connected devices.

The graph in Figure 1 shows timestamp information for a certain period, with the x-axis denoting popularity and the y-axis denoting fame beginning at 0 (minimum) on the way to 100 (maximum). The popularity of machine learning and cybersecurity expanded more
than twofold between 2016 and 2022. In this learning, we focus on pattern recognition in assaults since it is closely related to such areas as the level of protection, smart choice, and the information editing methods that will be employed in potential implementation. Machine learning methods are utilized to analyze cyber threats and enhance cybersecurity practices in this research. Investigators from both academia and industry that want to create data-driven, sophisticated cybersecurity algorithms using ML algorithms may also profit.

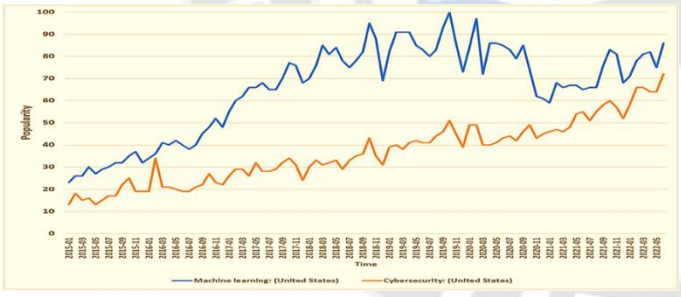

Figure 1: Google survey for security, data science, and deep forecast

Preventing cybersecurity assaults involves studying cybersecurity information and having the necessary tools to analyze it. Extracted features, linear regressions, unsupervised categorization, detecting interconnections, and neural network-focused reinforcement learning may be used to uncover vulnerability trends. This is covered in "Machine learning in cybercrime." With the use of these instructional methods, cyber-attacks might be avoided by identifying anomalies, malicious activity, and patterns in data-driven defense.

Deep learning (DL) is an incomplete but notable departure from conventional, well-known safekeeping products, such as firewalls, encryption systems, user authorization, and verification, which can or cannot be successful in addressing the demands of modern cyber businesses [16–18]. Domain experts and security researchers regularly update ad hoc information [19]. As more cybersecurity incidents surface, existing treatments have proven ineffective. As an outcome, a number of brand-new, sophisticated attacks appear and spread fast across the system. Academics use various data extraction strategies and information analysis to develop cyberspace strategies, which are discussed in "ML methodologies in cyber-security," based on the accurate identification of vulnerability insights and current safety patterns. According to research, cyber danger requires increasingly efficient and adaptive surveillance systems that really could react to attacks and adjust safekeeping rules to eradicate them. To do this, examining a sizable quantity of pertinent cybersecurity information obtained from many sources, including network and system sources, is necessary. Furthermore, these methods need to be used with little to no human involvement, increasing efficiency.

The following is a list of the study's key considerations.

- Research of the existing notion of cybercrime insurance products and associated methodology is first given in order to comprehend how data-driven decision-making may be applied. A range of machine learning (ML) methods used in cybersecurity is talked over, along with their utility and application inside this field.
- In our proposed ensemble technique, we examined a number of machine learning algorithms and employed both the best- and poor-performing versions to inspect the better results when the benchmark prototypes are being integrated.
- Our approach integrates the advantages and capabilities of various algorithms into a single, reliable model. By doing this, we ensure that the modeling we use to address the issue is in the best feasible solutions, enhancing the generality of our suspected cases.
- In order to validate our aggregation algorithm, we used three sets of data parameters.
- The suggested novel ensemble classifier improved generalization and outperformed comparable studies, as per analytical outcomes for the NSL-KDD, Credit Card Fake Detection, and UNSW sets of information. The rest of this research is divided into the next sections.

The purpose of our study and cyber-technologies are covered in Section 2. Then, it briefly examines a number of cybersecurity categories and subcategories, including information on cyber events. A brief summary of the main categories of machine learning (ML) methods and how they apply to certain cybersecurity complications are offered with

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5793
***Article Received**: 07 August 2022 **Revised**: 10 October 2022 **Accepted**: 06 November 2022*

brief elucidation in Section 3. The chapter also includes a list of several of the most in effect methods of ML for cyber-security models. Section 4 summaries and highlights the many outcomes in scientific concerns and future directions in cybercrime. Section 5 achieves the findings.

2. LITERATURE REVIEW

The infrastructure of tools for communication and information has considerably altered over the past 10 years, and it is now pervasive and fully integrated into our contemporary civilization. This section describes the cyber-attack techniques with respect to IoT systems based on machine and deep learning. Various ML & DL methods are discussed in this chapter as possible remedies to stop cyberattacks on IoT applications. A summary of the ML and DL algorithms utilized in IoT to identify cyberattacks is provided in Tables 1 and 2.

Table 1: Summary of predicting cyberattacks using ML

Approaches	Database	Function	Evaluation Metrics	Limitation
Semi-supervised ML [20]	MovieLens, bookCrossing, LastFM	Recommender System	The area under the curve	The suggested method's effectiveness is not shown
Various supervised ML [21]	Network activity data	IDS for Smart homes	precision, F-measure, recall	No overall accuracy standard is used.
Cognitive ML [22]	Information from trusted devices	Cyber-attack recognition in health care	Prediction ratio cost, delay accuracy, communication	The evaluation process wasn't clear.
ANN [23]	UNSW NB15	Cyber-attack recognition for smart cities	precision, f1-score, accuracy, recall	Techniques used to an incomplete dataset
ML [24]	MSRWCS	Cyber-attack recognition for multi-sources application	Accuracy	Insufficient validation metrics
ML with Fuzzy clustering [25]	UNSW-NB15	Cyber-attack on IoT Networks	Rate of Classification	Insufficient validation metrics
Semi-supervised algorithm [26]	NSL-KDD	IoT threat detection with distributed protection	Sensitivity, PPV, Accuracy	No test using actual world information

Malicious software may be transmitted if the attacker is aware of an application vulnerability, such as a SQL injection or false data injection, according to [20] coding that results. [21] created a three-layer IDS for smart homes using a classification algorithm. The three levels of the suggested IDS architecture work together to help the model identify malicious files. [22] suggested cognitive machine learning-assisted threat detection to protect health information (CML-ADF) framework. EML improved accuracy, attack detection, and performance above conventional approaches. An intrusion and anomaly wireless sensor were proposed by another study to spot vulnerability attacks in IoT-based intelligent city applications. The additional study provided probabilistic multi-model information for recommendation system security attacks [23]. The recommended technique beat previous models in recognizing recommendation systems anomalies. A linear classification iterate algorithm with excellent accuracy and low cost has been devised for the categorization of intrusions. [24] describes how researchers used an iterative simple linear classifier to identify intrusions and their sources. Although this was the case, there was a significant amount of incorrect categorization and it has to be reduced [25]. In contrast, [26] created an ESFCM approach using fog cloud technology to identify threats. This approach detects cyberattacks at the network edge and addresses distribution, scalability, and congestion [27], providing two level ensembles.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5793
***Article Received**: 07 August 2022 **Revised**: 10 October 2022 **Accepted**: 06 November 2022*

Table 2: Summary of predicting cyberattacks using DL

Methods	Database	Function	Evaluation Metrics	Limitation
Two-Level DT -Based DL and DNN [27]	Gas pipeline data from Mississippi State University and SWaT	Gas pipeline and water purification system cyberattack recognition and identification	Accuracy, F-score, recall, precision	high price of calculation
CNN [28]	UNSW-NB15	MCIDS	Accuracy and FP	No estimate Information
Key-Based Numeric Sequencing and the Fibonacci P-Sequence [29]	NSL-KDD	The water distribution system detects altered data	recall, F1 measure, accuracy, precision	Nothing is known about the shallow technique
DL [30]	NSL-KDD	Detecting attacks in social IoT	F2 score, precision, recall, F1 Score	Information is confined to one subject
Features are extracted using systemic NN and AE [31]	NSL-KDD	Detecting cyberattacks for cloud dew computers in automotive IoTs	Accuracy	For the Confirmation factor, insufficient
(CST-GR) [32]	BoT-IoT	Lightweight IoT system vulnerability detection	processing time	can only identify three types of attack
CNN's [33]	NSL-KDD	IoT environment vulnerability recognition and categorization	False Positive, True Positive, True Negative, and False Negative, K-fold cross-validation	There are no test results for practical methods

The framework for industrial control systems to identify and attribute attacks has been defined. First, deep representational learning detects control system imbalances; second, DNNs allocate assaults. [28] developed an MCIDS based on deep learning to recognize attacks using shellcode, generalist, worms, espionage, analysis, DoS, and fuzzes. [29] overcame the difficulty of changing data in the communications infrastructure that endangers cyber-physical systems. They protected the control system by encrypting output vectors with a key-based arithmetic sequence and Fibonacci p-sequences. [30] Recommended using deep learning (DL) to uncover unseen outlines in additional information in order to prevent social IoT assaults. They believe their technology can detect attacks more accurately than more traditional ML algorithms during information transportation between the cloud and automotive end-user devices [31] identified cyber-attacks. To recognize the assaults listed, they used a customized layered auto-encoder. A compact IDS predicated on the LMT, RF classifiers, J48, and the Hoefeding trees was built by [32] in another study (VFDT). They developed a state-of-the-art method known as correlated-set thresh-holding on the voltage increase (CST-GR), which only employs the attributes mandatory for separate hacks. [33] Created an IoT-based intrusion detection and classification platform using deep learning as well as a neural network model (IoT-IDCS-CNN). Feature extraction, segmentation techniques, and system identification compose the methods.

3. PROPOSED METHODOLOGY: Ensemble of Auto-Learning Classifiers

This section describes the presentation of an ensemble of tree-based classifiers in identifying Portable Executable (PE) malware. Our study of comparison is shown in Fig 2. Three

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5793
***Article Received**: 07 August 2022 **Revised**: 10 October 2022 **Accepted**: 06 November 2022*

different PE malware data communications are used to train several tree-based ensemble techniques, which then provide classification techniques. The performance of different classifiers is then evaluated by testing them against a test dataset. At last, a 2-stage statistical relevance test examines evaluation criteria. Next, we describe the malware databases and tree-based classification ensemble used in this investigation.

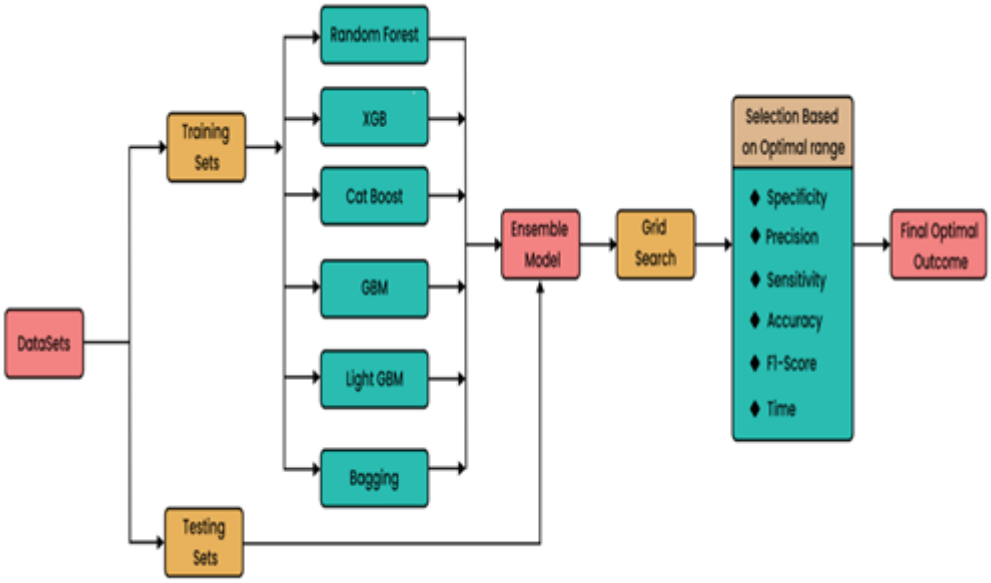

Figure 2: Tree-based ensembles analysis approach for detecting PEs

3.1 Set of databases involved

We used three distinct sets of data to train our models. The NSL-KDD and UNSW-NB15 datasets were used to train the ensemble method for cyber-attack detection. However, because no other sets of information had enough records of training a sophisticated model, the fraud-detecting ensemble model could only be skilled on a single dataset. Below is a discussion of every set of information that was used in this study.

3.1.1 NSL-KDD

The sets of information consist of network traffic records that a basic intrusion detection system has examined in the NSL-KDD set. These data represent the phantom traffic that an actual IDS would have seen. Each record in the collection comprises 42 properties, of which 2 are labels, and 40 are related to the traffic input. The first label specifies whether it is a typical occurrence or an assault, while the second describes the traffic input's seriousness. The NSL information sets are an enhancement to the KDD'99 sets, which had many duplicate records. The authors divided the NSL sets of information into test and training sets for the user's convenience. 11,285 records are in the test set, while 125,753 records are in the train set. Real network traffic statistics were gathered in 1999 as part of the Learning Discovery and Data Mining competition. Furthermore, the NSL test and train parameters have a respectable amount of data, allowing the tests to be run on the complete set without the need to select a small random sample. As a result, it will be simple and standard to compare the results of various research projects' assessments.

3.1.2 UNSW-NB15

The sets of information consist of fresh network packs produced by the University of New South Wales Lab using the IXIA Perfect Storm programme to provide a combination of genuine contemporary routine operations and synthetic contemporary attack behaviors as UNSW-NB15. It contained 100 GB of unprocessed traffic that was recorded with the TCP dump programme. 9 different forms of attack in this set of information are analysis, generic, fuzzers, DoS, backdoors, exploits, reconnaissance, shellcode, and worms. In this dataset, there are 2,540,078 records in total. 175,389 records from a subcategory of the information, which was utilized as the trained set, were included. A different subset was put up as the tested set, which had 82,345 records. Records reflecting both typical information and different types of attacks are included in these sets.

3.1.3 Set of data used to detect credit card fraud

This set of information includes credit card communications made in September 2013 by users throughout Europe. Out of 284,809 transactions over 2 days, there are 492 false records in this set of information. The dataset is extremely uneven, with duplicitous records making up just 0.192 percent of each

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5793
***Article Received**: 07 August 2022 **Revised**: 10 October 2022 **Accepted**: 06 November 2022*

transaction. Therefore, we have to pre-process the processes to balance the records between the 2
classes. This information was gathered as a result of a sizable research association between World line and the ML Group at the Université Libre on information mining and malware detection. The information was processed using PCA analysis due to concerns about data protection and contained just the numerical values of the primary components, with the exception of two columns (“Time” and "Amount") While the "Amount" column carries the transaction amount, the "Time" column displays the amount of time that has passed since the initial transaction, which might be useful for cost-sensitive research. Due to their sensitivity, the real attributes and transactional data were inaccessible, and all above are in [34].

3.2 Using a Tree-Based ET

Tree-based ensembles integrate fundamental learners (e.g., decision trees or CART) rather than creating separate readers from test examples. An ensemble is often created via two steps, namely the production of the basis learners and their integration. Generally, basic learning must be as precise and diverse as feasible for a quality ensemble [35]. This research takes into account four artificial neural algorithms based on trees. It is significant to note that each procedure's hyperparameters are tuned using a randomized research methodology [36].

3.2.1 Random Forest (RF)

RF is a tree-based ensemble whose trees rely on random factors. Breiman's [37] first description of the random forest technique is as follows. Random forest services trees $h_j(m, \Omega)$ as its base learners. Keep fit information D1 = $\{(m_1, n_1), \ldots, (m_\alpha, n_\alpha)\}$, where $x_i = (m_{i,1}, \ldots, m_{i,p})^T$ represents y_i represents the response and the p predictors and, and a specific manifestation ω_j of Ω_j, the fitted tree is given as $\widehat{h_j}$ (m, ω_j, D1). In Method 1, we see the details of how the random forest algorithm works.

We make use of a lightning-fast random forest implementation in R called rangers [38], which is designed for high-dimensional large datasets like the one we have. For each malware database, Table 3 lists the random forest's hyperparameters. The search space that we established for every hyper-parameter is termed as Split rule = "Gini", "extra trees"; number of trees = [50, 100, 250, 500, 750, 1000]; minimal network size = 1, 2,... 10, replacement = "TRUE, FALSE"; and mtry = number of characteristics = [0.05, 0.15, 0.25, 0.333, 0.4]; sample proportion =[0.5, 0.63, 0.8]. The first approach uses the standard randomized forest technique for classification techniques.

Table 3: The final metrics of learning spontaneously in a search algorithm

Hyper-parameter	Kaggle	BODMAS	CIC-MalMem-2022
Number of trees	1000	100	500
Mtry	30	119	18
Min node size	8	4	6
Split rule	‘Gini’	‘Gini’	‘extra trees’
Fraction of sample	0.81	0.64	0.64
Replaced	FALSE	FALSE	TRUE

Training procedure:

1. Necessitate - unique training dataset $\mathcal{D} = \{(m_1, n_1), (m_2, n_2), \ldots, (m_\alpha, n_\alpha)\}$, with $m_i = (m_{i,1}, \ldots, m_{i,p})^T$
2. for j = (1 to J)
3. Performed bootstraps instances $\mathcal{D}1_j$ of range α from $\mathcal{D}1$.
4. With a binary recursive divider, fit a tree on $\mathcal{D}1_j$.
5. End
   Evaluation:
6. Necessitate – For example, to the classifier x.
7. $\hat{f}(m) = \arg\max_y \sum_{j=1}^{J} I(\hat{h}_j(m) = n)$
   where $\hat{h}_j(m)$ represents the reply uneven at x using the j-th tree.

3.2.2 Decision Trees with Gradient Boosting

In this study, we also took into account other tree-based boosting ensemble spyware detection techniques, including XGBoost [39], CatBoost [40], GBM [41], and LightGBM [42]. GBDT ensembles are linear transformation models based on tree-based classifiers (e.g., CART).
Let $\mathcal{D} = \{(m_i, n_i) \mid i \in \{1, \ldots, \alpha\}\ m_i \in \mathbb{R}^\eta, n_i \in \mathbb{R}\}$ denote the malware database and compare η attributes & α illustrations. Consider a group of j trees, output $n(\hat{m})^j$ for input x is occurred by evaluating the forecast from every tree $n(\hat{m})^j$

$$n(\hat{\mathbf{m}})^j = \sum_{i=1}^{j} f_i(\mathbf{m}) \qquad (1)$$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5793
Article Received: 07 August 2022 Revised: 10 October 2022 Accepted: 06 November 2022

where f_i reflects the j-tree ensemble's i-th analysis tree's result. GBDTs reduce a normalized objective function Obj^t to construct the $(j + 1)$-th tree.

$$x\{Obj\ (f1)^t\} = x\{\Omega(f1)^t + \Theta(f1)^t\} \quad (2)$$

where $\Omega(f1)^t$ denotes loss of function, $\Theta(f1)^t$ is a normalized function to manage over-fitting. The defeat function $\Omega(f1)^t$ events show the disparity between the forecast $\hat{y}_i$ and the target y_i. On the other hand, the normalized role is represented as $\Theta(f1)^t = \gamma T + \frac{1}{2}\lambda \parallel w \parallel^2$where w and T denote weights in the tree and quantity of leaves in the tree.

3.2.3 XGBoost

A lot of progressively trained trees are produced by the scalable end-to-end tree-boosting approach known as XGBoost. [43] Each tree corrects the faults of the last, creating an effective categorization model. The generalization error problem of the method is solved by XGBoost, which also makes use of multi-threading with sparsity-aware measures. Two alternative XGBoost implementations, including native versions in R [44] and H2O [45], are used in this investigation. The hyperparameter search space for native XGBoost is configured as follows. Maximum depth is equal to {2,3,... 24, ...}, eta is equal to {0, 0.1, 0.2,..., 1.0}, subsample is equal to {0.5, 0.6, 0.7, 0.8} and column samples by tree is equal to {0.5, 0.6, 0.7, 0.8, 0.9}. Additionally, we defined the following search space for the XGBoost hyperparameters implemented in H2O. Maximum depth is {1, 3, 5,..., 29}; sample rate is {0.2, 0.3}; sample rate of column is{ 0.20, 0.210, 0.220}; column sample rate per tree is{ 0.2, 0.21, 0.22}; minimum rows are 0; and $\log_2$ number of rows-1. For both XGBoost methods, the final learning parameters can be found in Table 4.

Table 4: Randomly searching XGBoost's and final learning settings

Hyperparameter	Kaggle	BODMAS	CIC-MalMem-2022
Native			
Depth of Maximum	19	13	21
Eta	0.300	0.23	0.100
Subsampling	0.801	0.69	0.601
Column sample rate per tree	0.262	0.71	0.60
Column sample of tree	0.502	0.71	0.601
H2O			
Depth of Maximum	2302	24	26
Rate of Sample	0.950	0.53	0.981
Rate of Column sample	0.630	0.43	0.700
Rows of Minimum	2	2	2

3.2.4 CatBoost

Symmetric decision trees are the basis of CatBoost. It is acknowledged as a classification model that might provide exceptional outcomes and 10 times the predicted performance of methods that do not employ symmetrical tree structures. CatBoost can accept gradient bias and forecast shift to improve prediction accuracy and generalization of huge datasets. CatBoost consists of two techniques: sorted boosts, which anticipate leaf values throughout the tree-structured building to reduce the fitting problem, and a unique methodology for handling categorical variables during testing. Both of these approaches are included in the software. In this document, CatBoost is implemented in the examine space for each hyperparameter, and R is thought of in the following way. Depth = {1, 2,... 10, }learning rate = 0, learning type = "Ordered", "Plain," l2 leaves supervised = {3, 1, 5, 10, 100,} border count = {32, 5, 10, 20, 50, 100, 200}. Table 5 shows the final learning variables of CatBoost for every spyware data set.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5793
Article Received: 07 August 2022 Revised: 10 October 2022 Accepted: 06 November 2022

Table 5: Randomly chosen CatBoost learner settings

Hyper parameter	Kaggle	BODMAS	CIC-MalMem-2022
Depth	4	10	3
Rate of Learning	0.20	0.2	0.20
L2 leaf Normalization	4	5	6
Count of Border	101	100	50
Type of Boosting	"Plain"	"Plain"	"Ordered"

3.2.5 Gradient boosting machine

The first application of GBDT to make use of forwarding learning is GBM. Sequential tree generation occurs, with earlier trees' outcomes influencing later trees' generation. Correctly, GBM is accomplished by repeatedly building a set of functions$f^0, f^1, \ldots, f^t$, given a loss function $\Omega(y_i, f^t)$. We could optimize our estimates y_i by discovering additional functions $f^{t+1} = f^t + h^{t+1}(\mathbf{x})$to decrease the loss function's estimated rate. In this work, we implement GBM in H2O, and the search space is as follows. Highest depth = 1,3,5,7,...,29; sample rate = 0.20; column sampling frequency per tree = 0.20; column sample rate increase per level = 0.9; amount of bins = 2; 4,5,...,10; and minimal rows = {0; 1,...,$\log_2$; number of rows -1.} The complete set of final GBM hyper-parameters that was applied to each malware database is listed in Table 6.

Table 6: Final GBM training factors for every dataset after random verification

Hyperparameter	CIC-MalMem-2022	Kaggle	BODMAS
Highest depth	27	25	24
Rate	0.73	0.45	0.53
No. of bins	1024	512	64
Column rate change of -level	0.93	1.05	1.03
Column samples rate – tree	0.62	0.66	0.43
Lowest rows	8	2	2

3.2.6 LightGBM

LightGBM uses histogram and leaf-wise approaches to enhance computing power and model accuracy. It is possible to mix a characteristic that
are inconsistent with one another using the histogram approach. The fundamental concept is to discretize continuous characteristics into n integers before producing an n-width histogram. The training data is searched for the decision tree using the histogram's discretized values. The histogram technique greatly reduces runtime complexity. The leaflet in LightGBM that had the greatest splitter gain was found and separated one leaf at a time. A wider logistic regression and generalization error may result from leaf-wise improvement. LightGBM restricts leaf-wise depth in order to increase effectiveness and reduce the fitting problem. In this work, we used a LightGBM version in R with these hyperparameters: path smooth ={ 1 * 10-8 , 1 * 10-3}; Maximum bin = {100, 255,} maximum depth = {1, 2,..., 15} number of leaves = 2(1,2,...,15), minimum information in leaf = {100, 200,..., 1000} learning rate = {0.01 , 0.03 , 0.01} feature fraction ={ 0.5 0.9}; bagging fraction = {0.5 0.9};
path smooth ={ 1 * 10-8 , 1 * 10-3}; and minimal level gain to split = {0 ,1, 2,..} The complete list of final LightGBM hyper-parameters utilized for each spyware database can be seen in Table 7.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5793
***Article Received**: 07 August 2022 **Revised**: 10 October 2022 **Accepted**: 06 November 2022*

Table 7: LightGBM's complete learning parameters following the search strategy

Hyper-parameter	Kaggle	BODMAS	CIC-MalMem-2022
Bin of Maximum	100	100	255
Depth of Maximum	9	10	3
No. of leaves	8	8192	512
Minimum information in leaves	800	1000	700
Rate of Learning	0.28	0.29	0.07
Lambda l1	0	40	0
Lambda l2	20	90	90
Fraction of attributes	0.9	0.5	0.9
Fraction of Bagging	0.5	0.5	0.5
smooth Path	$1 \times 10{-}8$	0.001	0.001
Minimum gain to splits	15	2	11

3.2.7 Aggregating the Bagging

Bagging is a kind of machine learning that combines several techniques to improve their performance in statistic regression and classification tests. On the other hand, it also lowers variance and fixes several issues with overfitting the whole dataset. Because of this, the general algorithm for bagging is as follows.

Steps involved:

- "For m = 1/M/M" (iterations)

Drawing a sample from the S_m bootstrapping addressing the total data, studying just the C_m classifiers from the S_m, and providing examples for each test sample

a. Drawing the S_m random sample on the entire data
b. Learning about the C_m classification from the S_m

- Example of each test sample

a. Trying each of C_m's classifications

In contrast to the provided datasets, where each database is made up of "n" tuples via replacement, dataset D comprises "n" tuples and uses the method of bagging to generate "m." However, this sampling is also known as a bootstrap sample. As a result, each model is completely fitted with the random subset and completely merged by subtracting the averages of all the outcomes. Finally, the grid search model [45] depicts the findings' optimized outcomes.

4. EXPERIMENTAL RESULTS

Table 8 summarizes the output of using ensemble loading to find fraudulent credit card transactions. The research compared good and terrible machine-supervised learning. Repeated measures models were created, and 10-fold cross-validation was used to choose the best for zero level of the stacking ensembles strategy. Different databases assumed supervised learning. Rf, XGBoost, MLP, and Support Vector algorithms were best for fraud with credit card detection. In NSL-KDD and UNSW, the best ML techniques were RF, XGBoost, and Classifier. The training time was also computed for each model and ensemble stacking, as displayed in Tables 8–10. Figure 3 displays the NSL-KDD ROC curves, while Figures 4 show the UNSW ROC curves. Tables 8–10 show greater training durations for the best ML algorithms compared to the baseline models with poor performance. The ROC curve and accuracy were enhanced. However, the best approach for a given issue will depend on the situation. When time is most important, we may employ quicker but poor-performing ML algorithms, and when performance is most important, we can utilize top-performing machine learning (ML) procedures.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5793
***Article Received**: 07 August 2022 **Revised**: 10 October 2022 **Accepted**: 06 November 2022*

Table 8: Detecting credit card theft using ET stacking

Approaches	Specificity (%)	Sensitivity (%)	Precision (%)	Training Time (Sec)	Accuracy (%)	F1 Score (%)
Poor: Ensemble Stacking	0.964	0.912	0.969	8.43	0.935	0.939
ET Classification	1.001	0.828	1.001	8.35	0.907	0.907
Strong: Ensemble Stacking	0.964	0.904	0.969	21.72	0.931	0.935
Gaussian NB	0.984	0.859	0.988	0.09	0.917	0.918
RF Classification	0.999	0.867	0.993	3.07	0.923	0.924
MLP Classification	0.955	0.919	0.962	11.87	0.935	0.934
XGB Classification	0.937	0.913	0.947	1.38	0.923	0.929
Gradient Boosting Classification	0.946	0.898	0.953	2.2	0.919	0.924

Table 8 demonstrates that our stacking ensemble model outperforms all training sets and can detect credit card fraud accurately (93.5%). When we compare ensemble models with weak and strong base classifiers, we discover that both models implemented equally well, with the weak base ensemble learning outperforming the strong base prediction technique. As we move on to the stacking optimization strategy for assault prevention, we may look at Tables 8-10 to see how the models performed after being trained on various data sets. 20% of the NSL-KDD database was used to train the aggregation model, which outperformed the entire model (81.28%). This could be the result of generalizing using large datasets. Overfitting occurred when our machine learning algorithm attempted to cover too many data points. As noise and inaccurate data entered the model, efficiency and accuracy decreased. A machine learning model's generalizability is its power to adapt to unknowable inputs. This suggests that information analysis can produce reliable outcomes. The prototypical trained on the entire NSL database overfitted the training data and underperformed on the testing dataset; however, the model trained on only 20.0% of the data lasted well and correctly recognized attacks on the data. Compared to the NSL-KDD database, the accuracy of the UNSW-NB15 dataset is higher (95.15%) compared to (81.28%).

In general, the stacked optimization technique based on inadequate basis assumptions was more accurate than those with excellent base models. This may be meta-learners can acquire more from weak regular models than from strong ones, which are already correct. One experiment deviated from this trend.

Table 9: Ensemble stacking detects 20% of cyberattacks in NSL KDD

Approaches	Precision (%)	Specificity (%)	Sensitivity (%)	F1 Score (%)	Accuracy (%)	Trained Time (Sec)
Poor: Ensemble Stacking	0.805	0.718	0.885	0.843	0.813	37.96
RF Classification	0.878	0.871	0.709	0.784	0.779	4.6
ET Classification	0.966	0.973	0.572	0.719	0.746	14.34
Gradient Boosting Classification	0.967	79	0.628	0.758	0.798	1299
Ada Boost Classification	0.929	0.949	0.649	0.759	0.778	90.99
DT Classification	0.967	0.971	0.635	0.766	0.778	1.33
Gaussian NB	0.546	0.005	0.909	0.689	0.517	0.99
Strong: Ensemble Stacking	0.966	0.968	0.656	0.782	0.792	273.85

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5793
***Article Received**: 07 August 2022 **Revised**: 10 October 2022 **Accepted**: 06 November 2022*

Table 10: Using ensemble stacks, cyber-attack recognition for the NSL KDD database

Approaches	Sensitivity (%)	Precision (%)	F1 Score (%)	Accuracy (%)	Specificity (%)	Training Time (Sec)
Poor: Ensemble Stacking	0.6279	0.967	0.762	0.777	0.942	849.77
RF Classification	0.611	0.969	0.749	0.767	0.974	22.15
ET Classification	0.541	0.974	0.696	0.731	0.981	67.66
Strong: Ensemble Stacking	0.647	0.961	0.773	0.784	0.965	1669.05
Gaussian NB	0.039	0.949	0.088	0.458	0.999	0.79
DT Classification	0.649	0.968	0.778	0.787	0.974	8.72
XGB Classification	0.658	0.968	0.786	0.795	0.973	112.54
Gradient Boosting Classification	0.615	0.969	0.752	0.768	0.974	84.78

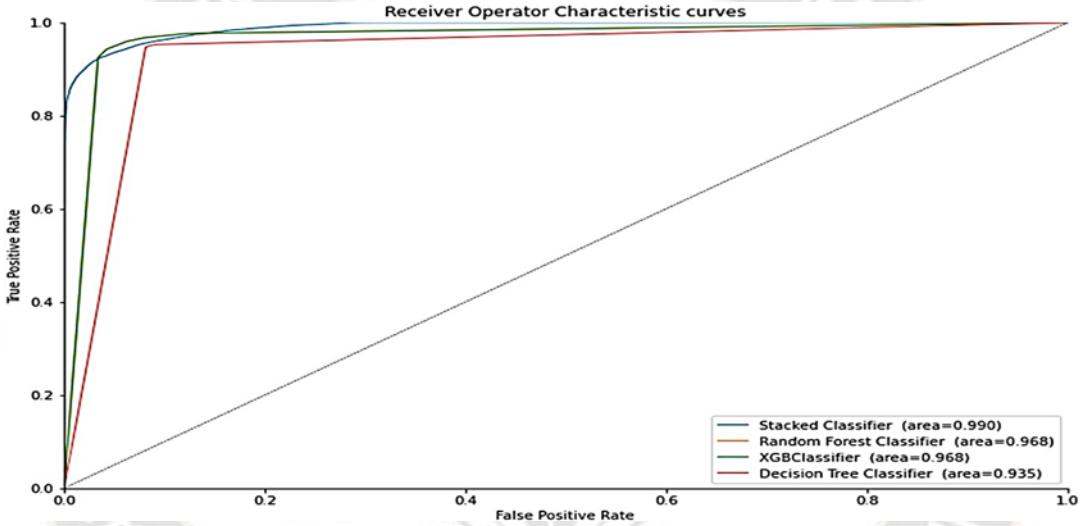

Figure 3: The NSL-KDD information sets the ROC curve

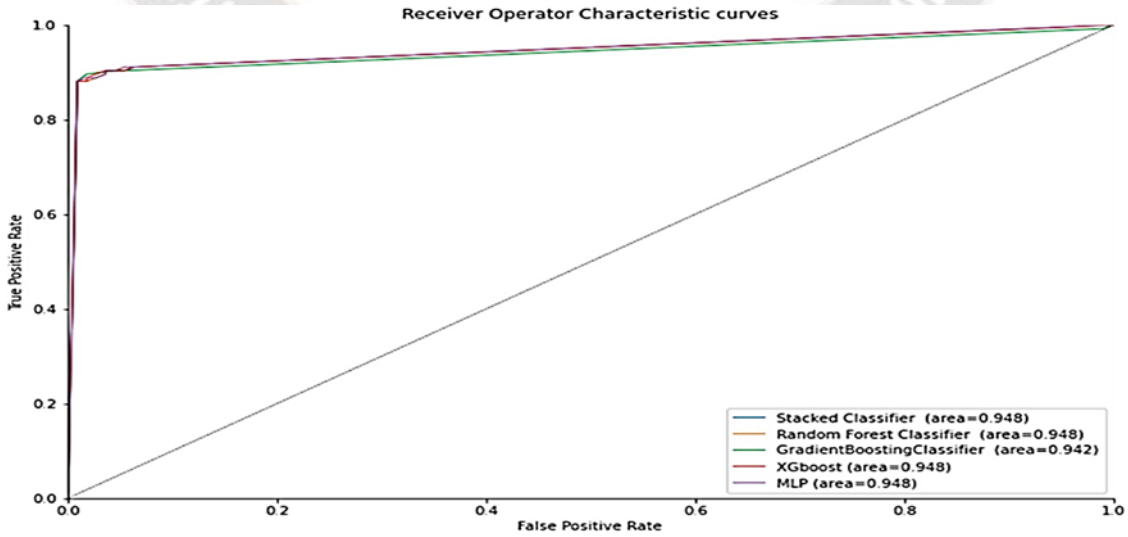

Figure 4: The ROC curve for a UNSW

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5793
***Article Received**: 07 August 2022 **Revised**: 10 October 2022 **Accepted**: 06 November 2022*

Table 9 shows that ensemble stacked algorithms with robust basis models perform marginally better. We noticed a similar trend with stacked prediction methods. All stacked concepts with weak base models had shorter trained times.

Furthermore, research revealed that the stacked prediction classifier's training time and the training time of its basis models were strongly correlated. Reduced base models trained faster than strong base modeling in each instance; hence the poor foundation stacked prediction method trained faster. Every stacking suggested approach had a Receiver operating characteristic area that was either larger or equal to all of its base models. Our stacked classification model outperformed other classification models.

5. CONCLUSION

The proposed ensemble algorithms used in this article to evaluate PE malware are tree-based structuring. A series of tree-based structured were similarly evaluated using numerous performance metrics, including various metrics like accurateness, precision, F1, recall, etc. These metrics included random forest, CatBoost, LightGBM, XGBoost, and GBM with bagging. We also included modern malware sets of data to understand the greatest recent assault trends. In addition to offering a statistical comparison of fine-tuned ensemble structures using various malware datasets, this work contributed significantly to previous research. The technology described in this research resolves the issue of detecting cyber-attacks in network traffic and can accurately identify fraudulent in transactions using credit cards. With a 95.15% accuracy rate and a training duration of 565.7 s, "ensemble stacking on UNSW-NB15" was our most appropriate prediction for cyberattack detection. Our "ensemble stacking (bad) strategy for credit card-based intrusion recognition" achieved similar results, with a precision of 93.9% and requiring only 8.5 s to train. Compared to most of the publications studied and described in Section 2, these findings demonstrate a significant advancement.

REFERENCES

[1] Li, S.; Da Xu, L.; Zhao, S. The internet of things: A survey. Inf. Syst. Front. 2015, 17, 243–259.

[2] Sun, N.; Zhang, J.; Rimba, P.; Gao, S.; Zhang, L.Y.; Xiang, Y. Data-driven cybersecurity incident prediction: A survey. IEEE Commun. Surv. Tutor. 2018, 21, 1744–1772.

[3] McIntosh, T.; Jang-Jaccard, J.; Watters, P.; Susnjak, T. The inadequacy of entropy-based ransomware detection. In Proceedings of the International Conference on Neural Information Processing, Sydney, Australia, 12–15 December 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 181–189.

[4] Alazab, M.; Venkatraman, S.;Watters, P.; Alazab, M. Zero-day malware detection based on supervised learning algorithms of API call signatures. In Proceedings of the Ninth Australasian Data Mining Conference (AusDM'11), Ballarat, Australia, 1–2 December 2011.

[5] Shaw, A. Data breach: From notification to prevention using PCI DSS. Colum. JL Soc. Probs. 2009, 43, 517.

[6] Gupta, B.B.; Tewari, A.; Jain, A.K.; Agrawal, D.P. Fighting against phishing attacks: State of the art and future challenges. Neural Comput. Appl. 2017, 28, 3629–3654.

[7] Geer, D.; Jardine, E.; Leverett, E. On market concentration and cybersecurity risk. J. Cyber Policy 2020, 5, 9–29.

[8] Buecker, A.; Borrett, M.; Lorenz, C.; Powers, C. Introducing the IBM Security Framework and IBM Security Blueprint to Realize Business-Driven Security; International Technical Support Organization: Riyadh, Saudi Arabia, 2010.

[9] Fischer, E.A. Cybersecurity Issues and Challenges: In Brief ; Library of Congress: Washington, DC, USA, 2014.

[10] Chernenko, E.; Demidov, O.; Lukyanov, F. Increasing International Cooperation in Cybersecurity and Adapting Cyber Norms; Council on Foreign Relations: New York, NY, USA, 2018.

[11] Papastergiou, S.; Mouratidis, H.; Kalogeraki, E.M. Cyber security incident handling, warning and response system for the european critical information infrastructures (cybersane). In Proceedings of the International Conference on Engineering Applications of Neural Networks, Crete, Greece, 24–26 May 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 476–487.

[12] O'Connell, M.E. Cyber security without cyber war. J. Confl. Secur. Law 2012, 17, 187–209.

[13] Tolle, K.M.; Tansley, D.S.W.; Hey, AJ The fourth paradigm: Data-intensive scientific discovery [point of view]. Proc. IEEE 2011, 99, 1334–1337.

[14] Benioff, M. Data, data everywhere: A special report on managing information (pp. 21–55). The Economist, 27 February 2010.

[15] Cost of Cyber Attacks vs. Cost of Cybersecurity in 2021|Sumo Logic. Available online: https://www.sumologic.com/blog/costof- cyber-attacks-vs-cost-of-cyber-security-in-2021/ (accessed on 10 May 2022).

[16] Anwar, S.; Mohamad Zain, J.; Zolkipli, M.F.; Inayat, Z.; Khan, S.; Anthony, B.; Chang, V. From intrusion detection to an intrusion response system: fundamentals, requirements, and future directions. Algorithms 2017, 10, 39.

[17] Mohammadi, S.; Mirvaziri, H.; Ghazizadeh-Ahsaee, M.; Karimipour, H. Cyber intrusion detection by combined feature selection algorithm. J. Inf. Secur. Appl. 2019, 44, 80–88.

[18] Tapiador, J.E.; Orfila, A.; Ribagorda, A.; Ramos, B. Key-recovery attacks on KIDS, a keyed anomaly detection system. IEEE Trans. Dependable Secur. Comput. 2013, 12, 312–325.

[19] Saxe, J.; Sanders, H. Malware Data Science: Attack Detection and Attribution; No Starch Press: San Francisco, CA, USA, 2018.

[20] Aktukmak, M.; Yilmaz, Y.; Uysal, I. Sequential Attack Detection in Recommender Systems; IEEE: Manhattan, NY, USA, 2021.

[21] Anthi, E.; Williams, L.; Slowinska, M.; Theodorakopoulos, G.; Burnap, P. A Supervised Intrusion Detection System for Smart Home IoT Devices. IEEE Internet Things J. 2019, 6, 9042–9053.

[22] AlZubi, A.A.; Al-Maitah, M.; Alarifi, A. Cyber-attack detection in healthcare using cyber-physical system and machine learning techniques. Soft Comput. 2021, 25, 12319–12332.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 1s
DOI: https://doi.org/10.17762/ijritcc.v10i1s.5793
***Article Received**: 07 August 2022 **Revised**: 10 October 2022 **Accepted**: 06 November 2022*

[23] Rashid, M.M.; Kamruzzaman, J.; Imam, T.; Kaisar, S.; Alam, M.J. Cyber attacks detection from smart city applications using artificial neural network. In Proceedings of the 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Gold Coast, Australia, 16–18 December 2020; pp. 1–6.

[24] Taheri, S.; Gondal, I.; Bagirov, A.; Harkness, G.; Brown, S.; Chi, C. Multisource cyber-attacks detection using machine learning. In Proceedings of the 2019 IEEE International Conference on Industrial Technology (ICIT), Melbourne, Australia, 13–15 February 2019; pp. 1167–1172.

[25] Cristiani, AL; Lieira, D.D.; Meneguette, R.I.; Camargo, H.A. A fuzzy intrusion detection system for identifying cyber-attacks on iot networks. In Proceedings of the 2020 IEEE Latin-American Conference on Communications (LATINCOM), Santo Domingo, Dominican Republic, 18–20 November 2020; pp. 1–6.

[26] Rathore, S.; Park, J.H. Semi-supervised learning based distributed attack detection framework for IoT. Appl. Soft Comput. 2018, 72, 79–89.

[27] Jahromi, A.N.; Karimipour, H.; Dehghantanha, A.; Choo, K.K.R. Toward detection and attribution of cyber-attacks in iot-enabled cyber-physical systems. IEEE Internet Things J. 2021, 8, 13712–13722.

[28] Singh, S.; Fernandes, S.V.; Padmanabha, V.; Rubini, P.E. Mcids-multi classifier intrusion detection system for iot cyber attack using deep learning algorithm. In Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 4–6 February 2021; pp. 354–360.

[29] Battisti, F.; Bernieri, G.; Carli, M.; Lopardo, M.; Pascucci, F. Detecting integrity attacks in iot-based cyber physical systems: A case study on hydra testbed. In Proceedings of the 2018 Global Internet of Things Summit (GIoTS), Bilbao, Spain, 4–7 June 2018; pp. 1–6.

[30] Preethi, P., & Asokan, R. (2019). An attempt to design improved and fool proof safe distribution of personal healthcare records for cloud computing. Mobile Networks and Applications, 24(6), 1755-1762.

[31] Moussa, M.M.; Alazzawi, L. Cyber attacks detection based on deep learning for cloud-dew computing in automotive iot applications. In Proceedings of the 2020 IEEE International Conference on Smart Cloud (SmartCloud), Washington, DC, USA, 6–8 November 2020; pp. 55–61.

[32] Soe, YN; Feng, Y.; Santosa, PI; Hartanto, R.; Sakurai, K. Towards a lightweight detection system for cyber attacks in the iot environment using corresponding features. Electronics 2020, 9, 144.

[33] Abu Al-Haija, Q.; Zein-Sabatto, S. An Efficient Deep-Learning-Based Detection and Classification System for Cyber-Attacks in IoT Communication Networks. Electronics 2020, 9, 2152.

[34] Yang, L.; Ciptadi, A.; Laziuk, I.; Ahmadzadeh, A.; Wang, G. BODMAS: An Open Dataset fr Learning based Temporal Analysis of PE Malware. In Proceedings of the IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 27 May 2021; pp. 78–84.

[35] Preethi, P., & Asokan, R. (2021). Modelling LSUTE: PKE Schemes for Safeguarding Electronic Healthcare Records Over Cloud Communication Environment. Wireless Personal Communications, 117(4), 2695-2711.

[36] Zhou, Z.H. Ensemble Methods: Foundations and Algorithms; CRC Press: Boca Raton, FL, USA, 2012.

[37] Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. J. Mach. Learn. Res. 2012, 13, 281–305.

[38] Breiman, L. Random forests. Mach. Learn. 2001, 45, 5–32.

[39] Wright, M.N.; Ziegler, A. Ranger: A fast implementation of random forests for high dimensional data in C++ and R. arXiv 2015, arXiv:1508.04409.

[40] Chen, T.; He, T.; Benesty, M.; Khotilovich, V.; Tang, Y.; Cho, H.; Chen, K. Xgboost: Extreme gradient boosting. In R Package Version 0.4–2; R Foundation for Statistical Computing: Vienna, Austria, 2015; Volume 1, pp. 1–4.

[41] Dorogush, A.V.; Ershov, V.; Gulin, A. CatBoost: gradient boosting with categorical features support. arXiv 2018, arXiv:1810.11363.

[42] Friedman, J.H. Greedy function approximation: A gradient boosting machine. Ann. Stat. 2001, 29, 1189–1232.

[43] Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. Adv. Neural Inf. Process. Syst. 2017, 30, 1–9.

[44] Chen, T.; He, T.; Benesty, M.; Khotilovich, V. Package 'xgboost'. R Version 2019, 90, 1–66.

[45] Cook, D. Practical Machine Learning with H2O: Powerful, Scalable Techniques for Deep Learning and AI; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2016.