NGEN Firewall Security Augmentation using Brooks-Iyengar and Random Forest Classfier method: by Predicting Cyber Threats from: Darkweb/Deepweb Data

Latesh Kumar KJ and Leena H U

There are many caveats persist in current Unified Threat Management Firewalls (UTM) device for assessing vulnerabilities in software and hardware. The techniques used to exploit threats in UTM appliances are less dynamic non-predictive and hence cybersecurity is still fickle. In this paper we propose a technique to integrate Brooks-Iyengar Fault algorithm and Random Forest Classifier model to analyze the dark web and deep web network analysis using machine learning methods with UTM devices to envision the exploitability of vulnerabilities. Our technique achieves this aim by analyzing vulnerability data from UTM logs, Microsoft and Redhat attack signatures, National Vulnerability Database and features established by users association with deepweb and darkweb (d2web) sites. We carry out a real-time experiment on a honey trap case study by involving real-time cyber criminals activity and vulnerability data to mitigate cyber risks. The results published are evaluated using F1 score and IPS and IDS improved by 16% while maintaining the performance and precision. We consider this result because many exploit cases recorded and documented of various vulnerabilities with their score are indicative of their ability in exposing the threat and impact, the prediction score by 94.3% shows the actual and subsequent threat analysis results with private cloud and elite firewall policy service.

Keywords: Unified Threat Management, Access Control List

1. INTRODUCTION

The cybersecurity community comprehends how global cyber-attacks are leveraged in exploiting the vulnerabilities. The cyber security venture sources dis-covered that every 14 second a new organization will fall victim of cyber-attack in 2019 and in 11 seconds by 2021. Another survey by Phishme discloses that 97% of ransomware attacks are increased in last two years because of increased digital computing. The Figure [1] shows the Ransom paid across major hit countries.

As a recent example, consider the new ransomware variant of WannaCry influences vulnerabilities in Taiwan Semiconductor Manufacturing Company (TSMC) in here, virus was spread highly in the secured facilities across 10,000 machines and manufacturer was forced to close the chip-fabrication. Microsoft has pushed an updated patch to newly discovered Wannacry Level vulnerability to secure Remote Desktop Protocol (RDP) and details are described in CVE-2019-0708. However, the new variant of WannaCry has disturbed large number of computers across the globe before Microsoft released a patch (CVE-2019-0708). The Common Vulnerability Scoring System (CVSS) [2019] has scored it 10 with other vulnerabilities, this highlights the existing CVSS scoring system is not quick enough in offering remedies. The running trends displays very less 3-5% of exploitation from the on registered vul-nerabilities [2019], [Carol Sabottke et al. 2015], [Luca Allodi et al. 2014], hence, to discover specific and dynamic vulnerabilities [Carol Sabottke et al. 2015], [Benjamin Bullough et al. 2017], [Mohammed Ali et al. 2018] new alternate proce-dures are implemented by cybersecurity researchers on social media and internet community blogs. The refer-ences indicate identification on influential users in significant propagation of information such

Dr. Latesh Kumar K J.,Technical Staff, Department of CSE, Siddaganga Institute of Technology, Tumkur-572103, Karnataka, INDIA, E-mail: latesh@sit.ac.in
Leena H.U., Full-time Research Scholar, Department of MCA, Siddaganga Institute of Technology, Tumkur-572103, Karnataka, INDIA, E-mail: leenahu@sit.ac.in

Acknowledgments: The paper was possible by contributions of SIT-ERP cloud data site Research Fund. The statements by author are solely the responsibility of the authors.

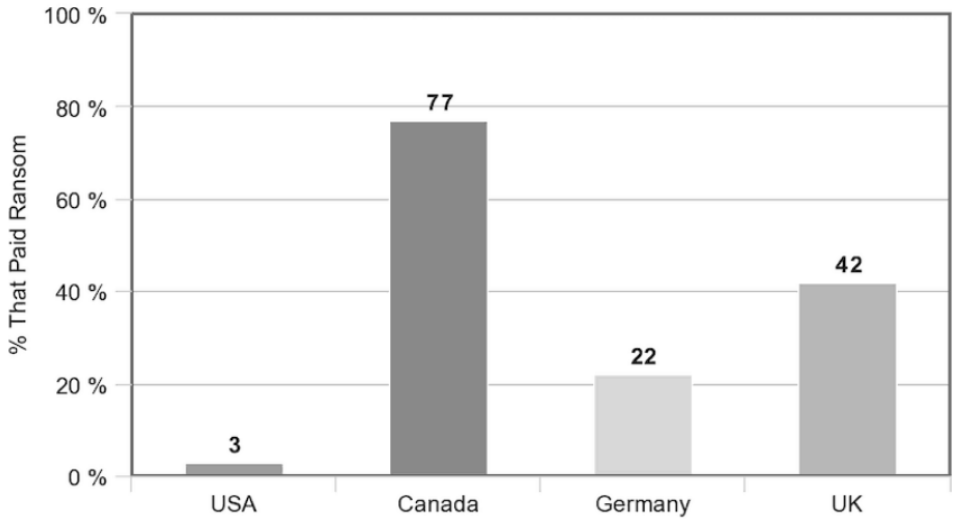

Figure 1.Countrywise Cyber Threats

as virus and bad activities across surface web and d2web. However, the users dynamics in connectivity, action, and behavior is not investigated legitimately to conclude as valid prediction. Furthermore, few more research studies are successful and recommend the study of d2web to crease seminal cyber risk information but, the real-time value and impact of intelligence that was responsible to predict a vulnerability is not disclosed with a quantifier.

We present a new dimension of reading and analyzing d2web data community blogs, firewall logs by Random Forect Classifier (RCF) and Brooks-Iyengar (BI) tech-niques with advanced Machine Learning (ML) to under-stand and predict the possible vulnerabilities of future by analyzing the vulnerability score. In this connection, we make use of CVE details of Microsoft, Sophos Firewall logs, CVE details from Red hat, CVE scores recorded by NIST and the leading Cisco Cybersecurity Corporation. In this paper, we present a case study demonstrating the ML processed d2web data for exploiting the counter measure for cyber risks. Additionally, we present a framework to showcase how firewall security device can dynamically integrate these smart intelligences as policies. We reveal how ML features interpret the d2web data for predicting the potential threats from harmful users in various meth-ods with the help of RCF and BI. The results are discussed by considering a private cloud and SOPHOS security device of the case study. Further, a peer comparision of result analysis is made on the impact of d2web social data on security policy procedures.

2. BACKGROUND

2.1 EVOLUTION OF VULNERABILITIES

In this subsection we provide a general description of the attacks and vulnerability lifecycles, the attack mechanism differs based on various dynamics of security practices (firewall) and vulnerabilities. We studied firewall security with OSI models and attack strategies based on the CVE repositories of Microsoft, Red Hat and NIST CVE data. Additionally, each vulnerability is evaluated using the ground facts obtained from attack signatures reported from cybersecurity firm, namely Cisco[1].

Inception of Vulnerability. The prime cause is overflow understanding of application security from its source by where vulnerabilities arrive. The four key sources of vul-nerabilities are 1) insecure coding practices, 2) non im-plementation of security practices in software/hardware development, 3) reuse of

vulnerable components and code and 4) idiosyncrasies of programming languages. The OWASP[2], is a worldwide non-profit organization focused on improving security of software. This organization provides supports with tool projects, code projects and documentation projects to manage the complete life cycle of software development.

Identification and Revealing of Vulnerability. The CVE Numbering Authorities (CNA[s]) creates repository by adding each suspected flow that exploits the security policy, each flow is validated for exploitability impishness and scanned for previous CVE ID to avoid replica. The vulnerability reported to CNA is verified and established by applying standard procedures. The CNA creates a Proof of Concept (PoC) to illustrate the exploitability of vulnerability with CVE details, this detail is not publicized till the vendor responds within the buffer time allotted for releasing the fix. The white hackers recreate the vulnerability with cited details to demonstrate the vulnerability, many researchers and software vendors opt to create PoCs for external world such as Rapid7[4] and further integrated with penetration testing Tools of Kali Linux5 for running experiments on various scales to provide remedies to vendors.

The National Institute of Standards and Technology[6] releases details to cyber security community about the vulnerability using National Vulnerability Database NVD[7] known to be ample data cite for vulnerabilities and disclosed over 8494 vulnerabilities from January 2019 to July 2019. Exploitation of Vulnerability. The NIST updates the vulnerability database all time and the increased number of vulnerabilities documented becomes the hurdle for many companies to diminish the activities of vulnerabilities and thus how exploitation widens. The current research and statistics disclose ground truth of vulnerability exploitation rating of 3% from the sources [Luca Allodi et al. 2014], [Michel Edkrantz et al. 2015], [Karthik Nayak et al. 2014], and [Carl Sabottke et al. 2015]. However, many exploits are spotted soon after the publicity. In a recent Risk Based Security (RBS) reports more than 22,000 vulnerabilities were disclosed in 2018 out of which 600 are unfixed, this is simply because many research studies have projected higher percentage of magnitude on lower risk vulnerabilities. Additionally, there are some high severity vulnerabilities are intentionally kept silent. In total, RBS had 6780 vulnerabilities in comparison with NVD database out of which 7% are higher severity are disclosed before the NVD database inclusion. It is evident that vulnerabilities scoring and reporting system are condemned for vulnerability information processing that results sometime false high rating on low rate risks. In another illustration a CVE-2014-0160 titled heartbleed was discovered roughly a day early after publication [Zakir Durumeric et al. 2014].

2.2 STEPPING INTO DARKWEB AND DEEEPWEB

The darkweb and deepweb are part of the darknet com-puting, d2web instigates from electronic forms like au-thentication, cash, hidden exchanges, anonymity, data and privacy [Moore D. et al. 2016]. The method of hiding the secured communication, the authentication plays vital role in securing the communication channel [Rivest R. et al. 1978] using RSA algorithm. The tactics of thumping the cash transactions using bitcoins [Cimpanu C. 2017]. The hidden services in the d2web passes information between intruders, basically this is achieved by hosting authenticated and secured website or discussion forums [Cimpanu C 2017]. The anonymity system concepts to be understood in thoroughly to access into d2web. The understanding of anonymity is not enough to venture in d2web, it requires the process of how to collect and process data from various sources. The open source web collection tool called web crawler is used to collect data. In general, the internet sources incorporates extensive data sets, the clear net and deep net layer is a subset of in-dexed search engine data. However the dark net data is not indexed by search engine in order to do we need specialized software tools like ToR: The Onion Router [Ciancaglini V. et al. 2015]. The support and encourage of criminals in sharing the data with a hidden layer is known as darknet anonymity, this is the ideal spot to recognize the future threats for society. The setup begins with installation of web crawler and scrapper system, the big system and natural language system (NLP). Further, the Linux operating system, anonymity service processing and elastic search tools is used fetch and analyse d2web data.

After configuring the base system, an intelligence mechanism must be triggered to read darknet data using language processor (OpenNLP). A gradle SKDMAN to be installed to support the processor and elasticsearch. Through this data entities like location, name, and date are captured through elasticsearch and using the open NLP processor it is analyzed with models. All these data entities are detected based on models that states what to detect from a defined model. There are many OPEN NLP pretrained packages [Reelson A. 2016] for reading data and its source from dark and deep web data. The figure [2] shows how openNLP Processor Testing on elasticsearch analyses the incoming source data. The integration of elastic search and kibana provides the model and data pattern analysis. The kibana searches log file, keyword data, and regular expressions/entities of NLP by performing various fil-tered analysis. The basic search of kibana on restricted data sets displays the real-world email, position to hacktivists, clues to exploits and real-time ip address and hostnames.

Figure 2. OpenNLP Processor

The next step is to fetch data on suspicious activities performed of the user/website using the web crawler by integrating Ahmia Scrapy Crwaler/Scrapper [Ahmia 2014] open source scripting tool that provides a strong darknet foundation for customization to fetch and process data based on model. The system extracts data from the darknet URLs through scrapy services of Ahmia, the data is mined by consuming the TOR services and dumps the extracted data into NLP processor by using the elasticsearch. In order to store such large volume of data Ahmia crawler requires a local database instance in the setup. This required setup was created on the case study suite using VMware services.

3. METHOD

In the proposed method we have opted to predict the vulnerability exploitation using the Random Forest Classifier (RFC) technique based on machine learning. To mitigate the faults of prediction we have integrated intelligence of Brooks Iyengar (BI) Fault Tolerance [Richard Brooks et al. 1996] algorithm. In this approach as name infers, this creates large number of vulnerability data operated as an ensemble. Each suspicious matching data is spit out by random forest and the data with the most certainty in match to NVD score is considered to be the prediction result. The proposed approach relies on supervised machine

learning data of d2web data and features derived from NVD. The data and features are evaluated with prediction based security appliances using the fault-tolerance system. The distributed decision making and precision accuracy is achived by merging the work of Mahaney and Schnedider [S. Mahaney et al. 1985] on Fast Convergence Algorithm (FCA) with the optimal regional algorithm. This algorithm is conceptualized based on Brooks-Iyengars multidimensional algorithm [Richard et al. 1996]. The algorithm error prediction capabilities on machine learned data is narrated below and work trails. The algorithm work follows as described in Table[1].

Algorithm 1 Prediction Error

Input:
The data sent by Machine Learned K where 1 ¡= K ¡=, and the prediction received from K can be denoted as [lk, hk]. Let z be the faulty predictions.
Output:
As proposed, prediction output is a point estimate and measurement of accuracy.
1. Machine Learned system receives point esti-mate and corresponding predicted data sets from sources (Sophos Firewall 1Logs, NVD Score, D2web social community posting data and Redhat/Microsoft Signature Data).
2. Take the union of predicted collection datasets.
3. Divide the union equally sole prediction object based on the data source and the accuracy level. We name the number of intersected prediction accuracy on random data sets for the particular source of target.
4. Identify prediction with accuracy less than N -z. Let N -z be identified as Faulty(F).
5. The sey of balance machine learned data S = (A_1, P_1)....(A_1, P_1) where A_1 and P_1 denote to accuracy and weight for the i^{th} prediction data corespondingly.
6. The prediction accuracy can be comnputed as:

$$P_{\mathrm{a}} = \frac{\frac{\sum_{A(A_{\mathrm{i}}+h_{\mathrm{i}})}}{2} * P_{\mathrm{i}}}{\sum_{F*P_{\mathrm{i}}}} \tag{1}$$

7. The prediction accuracy is evulated as (l_{i1}, h_{i1})

Table I: Brooks-Iyengar Algoritm Prediction Accuracy

Given a data set of n sources with z prediction objects, each prediction data set presents its core constructed features, the algorithm output comprises the near functionality and a corresponding threat counter system. In this system the input data for prediction is supplied to RCF based machine learning process and the output received from this system is supplied to the BI algorithm. The algorithm is integrated to achieve prediction accuracy and precision from hybrid data sources, even in the presence of faulty and fake data source. The algorithm does this by exchanging the predicted and accuracy data at every stage with every Red hat/Microsoft signature data, NVD data and d2web social community posting data. In addition, it is used to evaluate the accuracy of the predicted data for the whole network from all of the data collected. The algorithm demonstrated that even if some of the data from the source is faulty, the prediction network does not malfunction. The seminal thought in integration of BI algorithm is to analyse the data generated from the machine learning on UTM logs, user activities on social media and data posts on websites. The algorithm helps to identify the faulty from wild. The Figure [3] gives basic understanding and overview of proposed prediction of vulnerabilities from d2web data.

The architecture is categorized to 4 sections, input, feature extraction using machine reading, threat prediction and proactive security patch building by using BI and RCF techniques. In the illustrated architecture we have considered SOPHOS XG firewall for operation and evaluation. The firewall logs are imported to MSSQL database by granting permission within the Sophos Logwriter schema. This settings involved the account mapping with SQL login which is allowed execute SELECT and EXECUTE permission on log data. This secured external database setup was essential to store data in various meta-structures for analyzing the risk and environment based threat systems of firewall configuration and additionally, we

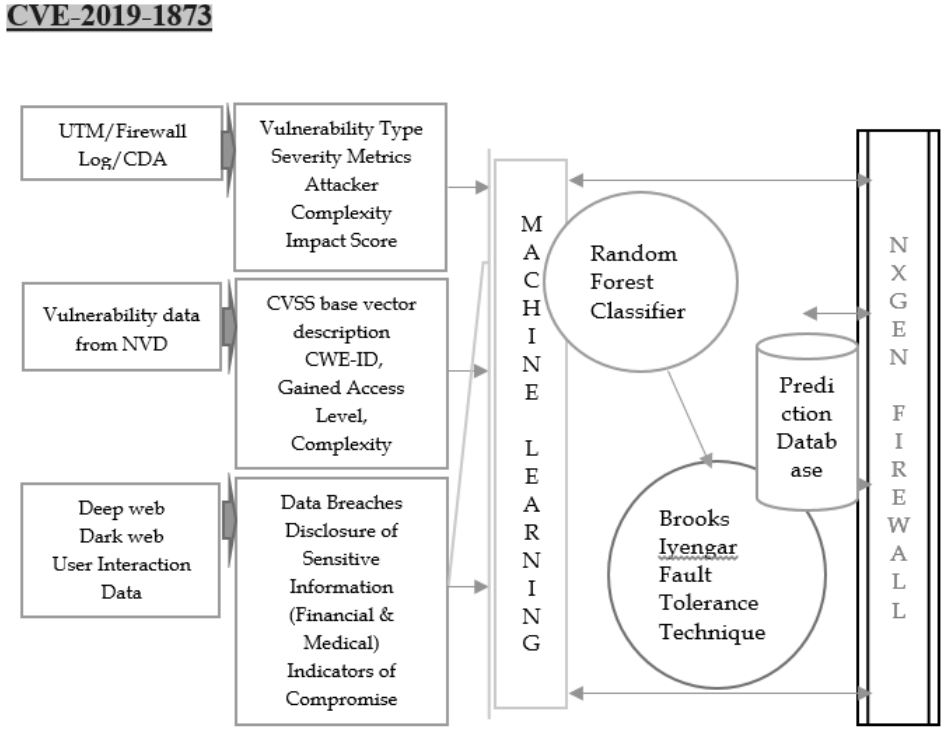

Figure 3. NGEN Prediction Model Architecture

obtained data from web crawlers to store using the web POST methods onto SQL data sets. The collective data from d2web posting, firewall logs, and NVD CSV data is stored at a centralized encrypted database system. Few key log types used in the proposed system is listed in the Table [2].

Name	**Description**
ErrorLog	Error log messages for all services
SMTP Error	Mail Transfer Agent Proxy Service errors
SMTP Panic	Mail Transfer Agent Proxy Service panic
IPS	Intrusion Prevention filter service
Application Filter	The application filter uses the same service and logfile as IPS
IDS	Intrusion Detection filter service
Blocked IP	Blocked Internet Protocol address
Blocked URL	Blocked www repository names
UImpact	Users impact data on internet

Table II: Firewall Log and Descriptions

The proposed setup involves 2 firewall devices connected in two different work environments locations, the configuration and policies defined for threat protection also differed based on requirements. //

3.1 DATA COLLECTION

The various features are generated through the data sources, UTM firewall logs, d2web database [Eric Nunes et al. 2016], NVD vulnerability CVSS data, Microsoft [2019] and Red hat sig-nature data by the proposed hybrid integrated model. In specific the UTM and d2web data is filtered on defined feature extractions for analyzing the cyber threats. The anonymity data collected from NVD repository, UTM logs, d2web related to vulnerability and exploitation from various web sources is stored in customized

MSSQL database. The core objects are discovered on the grounds of history of attacks and exploits discovered in the past from the UTM logs and Microsoft/Red hat vulnerability report. The study primarily aims on Vulnerability data posted by NVD and Microsoft security intelligence report data from January 2014 to December 2018. Figure 4 illustrates the number of susceptibilities circulated on NVD, Microsoft and Sophos firewall, this explains the how these vulnerability reporting is performed by each contender in differed time lines.

Sophos UTM: The Sophos XG 550 firewall data is extracted in CSV format on categories like User Threat Quotient (UTQ), Intrusion Attacks (IA), Intrusion Source (IS), Advanced Threat Protection (ATP), Security Heartbeat (SH), and Sandstorm that is a unique feature of firewall to diagnose threats. In addition, we also extracted data on Email Protection, SPAM, Virus Summary, Traffic, Security, Policies and few executive records. Each category of log record extracted is processed using the Random Forest Classifier and BI influenced machine learned methods. This data is uploaded to database using customized software tool, this software tool is developed to process various patterns and features based on Demand Filters (ODF) from which combinational features are generated.
Microsoft Threat Report: We mark vulnerabilities of matching the CVE-ID in the table description of the So-phos UTM with Microsoft and Redhat threat report (CVE-ID), the threat statistics are marginally less compared over Microsoft services and hence we focus on maximum scores of vulnerabilities published by Microsoft and Sophos, the reports of the previous works [M Almukaynizi et al. 2017], [E Nunes et al. 2016], [M Almukaynizi et al. 2017] focused on specific services based on CVE ID.

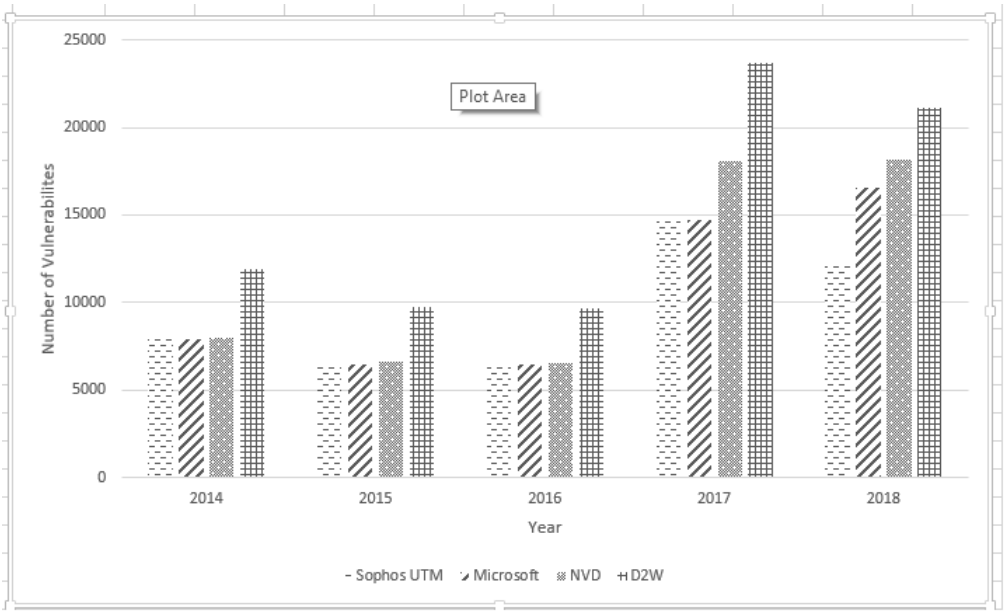

Figure 4. Year wise number of vulnerabilities disclosed by Sophos UTM firewall, Microsoft. The epxloited vulnearbilities reported by NVD and vulnerabilities mentioned on deepweb and darkweb

D2web Forums: The data extraction from deweb is tedi-ous as anonymity of website is highly maintained and integrity of the information extracted must be authenti-cated. There are plenty of websites

available when browsed through TOR browser and we used Kibana open source tool for crawling numerous websites that ends with .tor and onion extension on unlimited domains including (game station forums, quick money making sites, hitman, revenge forums, pornography and drugs). The search omitted large volume of result related to d2web posts, but key challenge was to filter hacking/attacking based data. This created a bunch of hacker decision tree from the large search results and then aggregated the votes from different hacker data tree to conclude the final data object. The kibana tool crawled 4, 35,400 web posts from which 231 unique forums are listed. From these data sets, we found genuine total 8762 exact match of vulnerability references matching the CVE-IDs of Sophos, Microsoft and Redhat.
NVD: The total vulnerability reported by NVD between the years 2014 - 2018 is 57387, this data set is read through the JSON script by matching each CVE ID referenced by NVD on specific threat report. In the sub section 3.3 the discussion of NVD features against So-phos/Microsoft/Redhat threat report is listed in detail. The study focused on evaluating the data and metadata description of threats and vulnerabilities.

3.2 PROPOSED PREDICTION METHOD

We applied Random Forest Classifier method to train the data using classifiers, Import Library, Create Model, Train and Predict. The study focused to extract features from Microsoft and NVD match the d2web posts aiming to breach Sophos firewall system. We used RCF for feature selection and recommendation engine. In section 3.1 we have described how data in collected, further the RCF selection technique makes a list of categorized threats based on input tree data set, then using this data set from each tree it recommends the most voted data set from the tree matching the pattern from all tree data sources of NVD, Microsoft, Redhat, Sophos and d2web. In this decision system there are two fragments, getting all the threats matching data from all the trees (tables) and generating one recommendation from each of the tree data source. Finally, a voting of each recommendation is done to finalize the best prediction data set. The individual vulnerability data tree is generated using an attribute selection indicator like, resource, threat, attack, impact, past action, action type and environment parameter. In addition, the system basically categorizes data majorly into two kinds such as exploited and non-exploited data. This separates the data sets of RCF tree for each instance and from which feature selection is made. The Figure [5] illustrates how the prediction of threats is selected based on vulnerabilities data sources. All data volumes from four defined sources are fetched into single pool of SQL data source with various filter levels for extraction.

Each data set is trained based on the filter levels of the input data source and each data set is passed to decision tree constructed using python class random Forest for random selection and evaluation. The shortlisted data set is now again voted by the trained data set and the input source counters in order to select as final predictive output. The final predited data set is recognized as set of nodes and each node is applied the BI intelligence to identify the faulty/fake node from the group of data nodes of the random decision tree.

3.3 PREDICTION MODEL

In this section we are decoding the Random Forests for threat data prediction of real-world machine learning problems. The Random Forest algorithm is used with minimal data cleaning for selecting random data set to list the features and decide the final result. In this paper, we have used random forest ensembles to function as divide-and-conquer tactic to increase performance of sole pathetic threat decision tree data models. The core objective of this is to understand how weak data sets are generated from strong data sets. The figure [6] shows how random forests are created using the python code for selecting the features based on data and shortlisting of the final decision tree data.

Description: We implemented it by using Python code base involving the Object oriented Programming method by defining a class Random Forest and Decision Tree. Each class consists of random forest and decision tree objects and member functions to serve as collector and average calculator for each of the

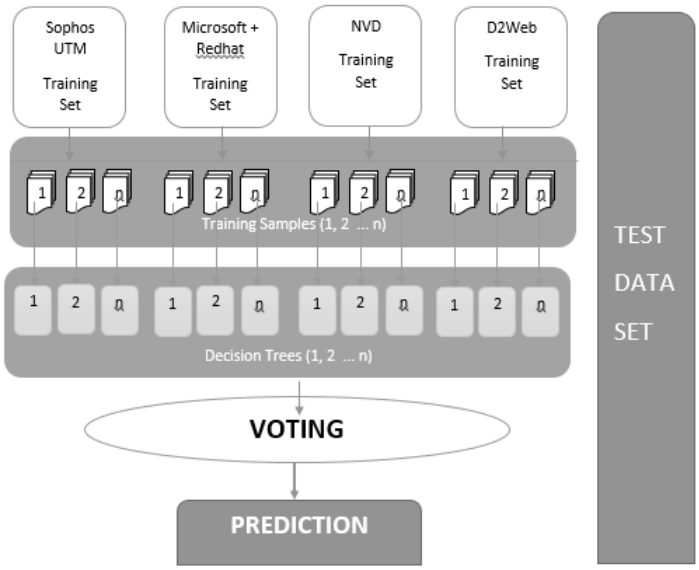

Figure 5. Proposed Threat Predicion Model - Machine

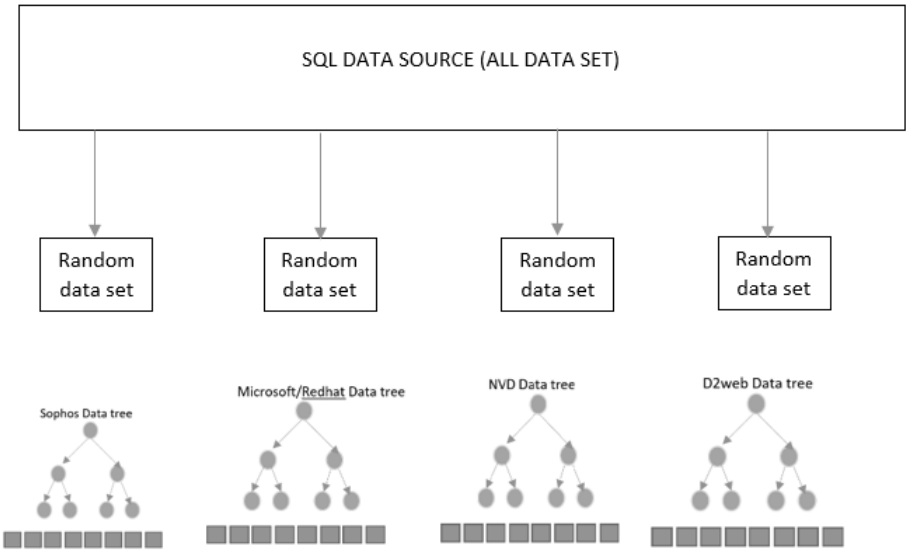

Figure 6. Random Forest Decision Tree Data Models for Threat Predicion Model

core functions performed by its Decision Trees. The first test suite started with 10 trees, 5 samples leaf and maximum depth as 4. The random forest class invokes 3 member routines Create_tree () to create base tree, Fit () to split the methods of underlying decision tress, and finally the Predict () to do the predictions from each tree. The decision tree consists of 9 routines, the first is initializing of decision tree

using Init_tree (), the Is_Leaf () is used to equal score the tree, Find_Better_Split () is used to get the best feature by splitting the tree to get a best, the Predict_Row () is used to envisage a single row data. The study focused only on two vulnerability types listed in the Table [3] and obtained data feeds from four data sources, this data is uploaded to customized SQL data source using the equation 2 for data processing.

$$Create\_tree = load\_data() \tag{2}$$

The test data generated between the years 2010 to 2018 is uploaded using function load_data (). The data sources named such like DS1 as: Sophos, DS2 as Mi-crosoft/Redhat, DS3 as NVD and DS4 as D2web.

Vulnerability	**DS1**	**DS2**	**DS3**	**DS4**
SQL Injection	1	1	3	5
Port infiltration	5	2	5	7

Table III: Each Random Tree Data

$$Create\_tree = fit() \tag{3}$$

In the below Table [4] we added dynamically a column of data called as Predict Data Pattern (PDP) by which we predict the vulnerability of the Table [3] data using the equation (3) and the routine fit ().

Vulnerability	**PDP**
SQL Injection	Username, Get Parameter
Port infiltration	Listening TCP/ UDP

Table IV: Each Random Tree Data

Now we created a training and test data based on Table [2] prediction data pattern. The program creates new data column for each PDP row matching the data attribute and it is marked as True else false by generating 0 and 1. The trained data equation is configured to analyze the random data tree objects by setting the accuracy length ¡=.75, this is a quick way to randomly evaluate the data patterns in shortlisting the final prediction set.

$$Data\_list = predict(test(features)) \tag{4}$$

The test and train data loaded in the database from each random tree is processed using the decision tree class which contains the predict function for shortlisting the final data set by matching the features extracted from each PDP and vulnerability type data row. For an illustration from the test Table [3] and Table [4] data we have obtained the below prediction data sets.

Vulnerability	**PDP**	**Final Prediction Count**
SQL Injection	Username, Get Parameter	3
Port infiltration	Listening TCP/ UDP	7

Table V: Prediction Data Set

3.4 FIREWALL AUGMENTATION

In this subsection, we discuss two core objectives, port infiltration and sql injection in a detailed description and provide gap analysis of firewall and how the study focused to augment the new methods to counter the port infiltration and sql injection threats from known and unknown sources all the time at the firewall. To assess the port infiltration we used nmap [2019] open source tool, this highly benefited in discovering data and metadata about port, hosts, and scan information of network and servers. The case study server is installed with this nmap server and a shell script is created and scheduled to run using the crontab service of Kali Linux for every 30 seconds and data collected by script is stored to a flat text file, after every day this file is uploaded to sql database for detailed analysis with machine learning methods. Figure [7] shows detailed listing of port scanning details of the honey trap server for cyber criminals. This server was setup with digital payment services and high volume of data interaction with external world. The port scan report continuously reported more than 10 port vulnerability repeatedly in spite of increasing the firewall policy rules when this was controlled by the cyber criminals with customized software tool. We kept this activity running for 3 weeks and every time when we fix the issue and restored the server using VM cloning method, in few hours the server was attacked using the other open ports with a different strategy. The script created by us captured the complete foot prints of each new attack by the cyber criminals on various parameters like host, port, trace methods, signatures and their dynamic internet protocol address from different locations. Using these dynamic IP addresses we discovered their random attack location and paths from of different geographical location and patterns of attack by mining the d2web data sources on port scan-ning data patterns.

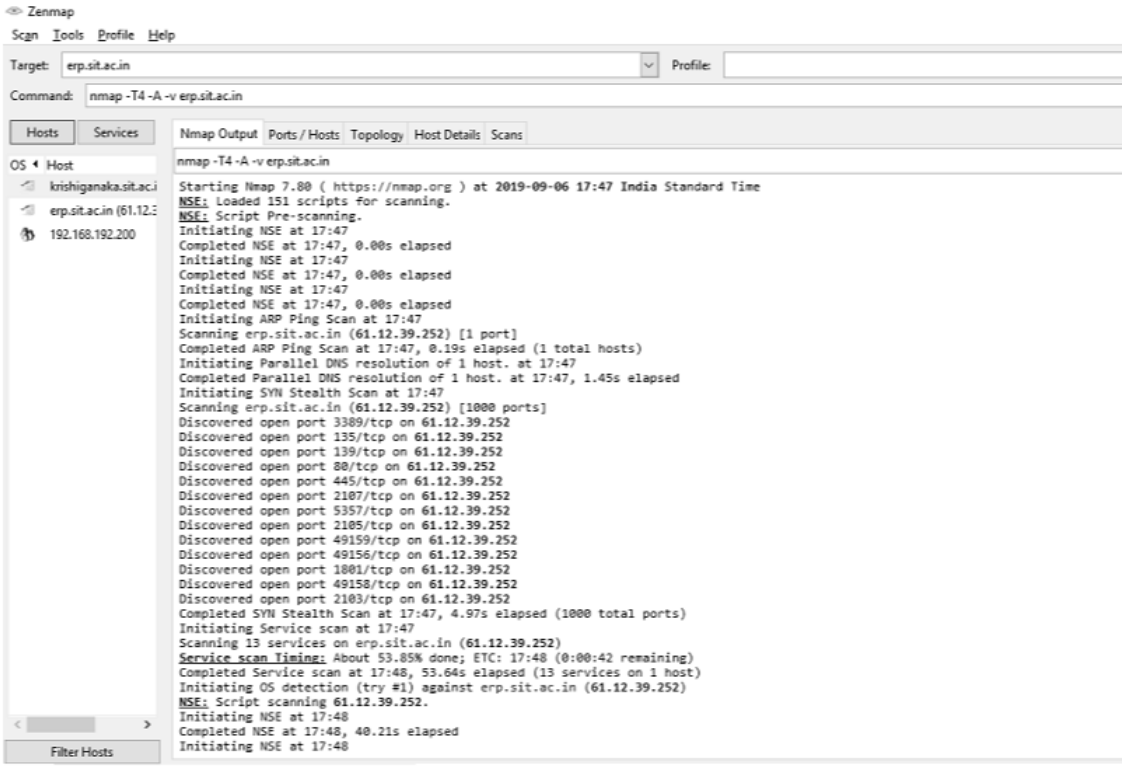

Figure 7. nmap port scanning of case study server through firewall service

The attackers entered our server using android Applica-tion Programming Interface (API) connecting the web services which interacts with payment detail database. At a point they succeeded invading the

single database source and took control over the database using the out-of-band [Yakkala V. et al. 2012] sql injection technique. The each client web query parsed is stored to encrypted hidden log file and this file was untouched by the criminals. The below listed databases were installed to find out most possible foot print attacking methods to discover the database types by attackers in wild. We created scripts on sql injection related data and stored them into flat text file and each error message is compared when an application is authenticated for access through the firewall data set. For an illustration to find out the type of the backend database an attacker injects the queries like listed below in Table [6].

Query	**Select eid,ename from eusers where eid=10 UNION select 1, version () limit 1,1**
Oracle	ORA-00933: SQL command not properly ended
MS Sql	Microsoft SQL Native Client error 0e148004 unclosed quotation after character string
MS Sql	an error in your SQL syntax; check the manual that matches to MySQL server version correct syntax to use at line 1 '\'
PostgreSQL	Query failed: error in syntax at or near The character "" 65 in /var/www/erp/test.php on line 4021.

Table VI: Footprints of Tactics

The different versions of database installed on various virtual machines are controlled by various firewall policy and when we recreated the attack foot prints of the criminals we noticed that the error messages matched theme of message except the line number and error numbers. The degree of attack and their incoming and outgoing traffic found to be same except the identities of the IP details. This firewall experiment resulted large volume of data for machine learning approach.

4. EXPERIMENTAL OPERATION

The experiments are conducted on vulnerability data set recorded from 2014-2018 specific to port infiltration and sql injection. The study focused only on two parameters and excluded all other data obtained from various d2web sources, each source data is verified with the data listed by NVD, Microsoft and Sophos firewall and no matching data is ignored in order to achieve the precision. We found more than hundreds of repeated vulnerability disclosures by various sourced through d2web traces and we ignored these data sources to maintain the genuinity of vulnerability data. Our concluding result dataset on vulnerability contains 231 specific to port infiltration and sql injection during the period of 2014- 2018, which have the class label exploited.

We examined machine learning classifiers like nearest neighbor and linear classifiers on our experimental setup data, but recorded results of random forest classifiers with brooks iyengar algorithm. The data created by random decision trees of RCF is picked randomly and trained by the brooks iyengar train the faulty node/data achieved better performance and varies from the native fault tolerance algorithms. The said fault tolerance system proved to be strong in identifying the fake/faulty data from the pool of vulnerability data list of the SQL database. We implemented 4 random decision trees on each data set to select the feature, this was further split into two record sets, success and failure based on prediction results. The impact of exploiting the vulnerability is based on score and predicted patterns are set with a decision limit. Ecah category of records are recognized as nodes, each of these nodes are verified for their fake identity of information.

4.1 EXPERIMENT TECHNIQUES

We conducted three experiments on two different firewall devices such as Sophos XG -250 and Next Generation Firewall (NGF Gaj). The first experiment focused on techniques engaged to bypass the firewall

by attackers, compromising the most vulnerable ports, second targeted the user involvement in various malicious d2web activities, and third focused to setting up a honey trap for cyber criminals. All three experiments involved a private mimicked commercial cloud information and this attracted the cyber criminals with suitable network traffic. The analysis of machine learned methods on Sophos firewall log data discovered major clue from the data patterns. The key observation extracted from random decision tree based on the data sets like port, scan, incoming, outgoing and Access Control List (ACL) policy is that, the ports of DNS 53 and FTP 20 are always kept open and no incoming traffic data pattern is found from the decision tree analysis. These ports falls into trap of the attackers as they are kept open. We involved this weakness in our second technique to observe how it works for cyber criminals. In the second technique we crated virtual machine to run digital payment service running the customized software application with MSSQL database service. This VM is assigned a public IP address and resolution is created in the DNS server and an FTP access is created to upload files through software application that communicates directly to database. The incoming ports DNS 53 and FTP 20 were opened, the smoke was created in various web sites using the URL fakewish.com on attractive large free fund schemes using d2websites. We noticed, most attacking techniques involve sensitive routine common weakness and once they fail, then attackers continue the effort in deep such like ignorant areas of administrators and environments. In this effort the DNS and FTPs incoming port was targeted by attackers to perform Fast-flux DNS attack by swapping extremely the DNS records in and out to forward the client request to fail.

In the third technique we opened common vulnerable ports in the VM machine and performed lots of malicious downloads from various onion and tor sites to invite the attackers interest on this setup. The setup was successful only once to get the attention and action from Bitcoin ransomware criminals. The attackers used our Remote Desktop (RDP) 3389 and DNS port, they took control over database and operating system by renaming all the files .locked file extensions and web server of the VM was killed, the database engine was killed and files related to database were encrypted with a lock. A customized windows application was installed in the windows registry with boot effect. We attempted many reboots with various settings and removal techniques but no joy in fixing up the issue, we used RSA and DES based key techniques to remove the encryption key from the source but it didnt work. All our effort in restoring the server back to operation failed, but the only one hand of success we has is, the hidden file that was kept in getting each execution was safe and unlocked we used this flat text file and this file helped us analyzing the navigation of attackers with the VM machine.

The technique captured attackers continuous port scanning data, from which the investigation on PS_LIMIT of IP address surpassing the limit with PS_INTERVAL values were measured and found there were more than 70 different IP addresses used to reach the port from various geo locations. Further, the port knocking was set to 0 and later we updated this parameter to 1 to receive continuous alerts whenever ports are knocked. Another method port spoof employed in the firewall generated useful data on open ports and their services with signature of databases.

4.2 EVALUATION OF PERFORMANCE

The proposed technique is evaluated using the F1 score and confusion matrix method, the counter, precision, recall and the feature related data of counter and recall are used to evaluate the performance of models [First. Last Accessed 2018]. The counter is the fraction of total security patch created based on predicted vulnerabilities, and precision is considered to be the actual exploited and derived from total predicted vulnerabilities. Table [7] explains the computation of each parameter and expression

5. RESULT DISCUSSION

In this subsection we discuss on two sets of experiments conducted and the obtained results. The experiments are conducted by setting the basic evaluating parameter as exposed vulnerability data and CVE score. We collected 231 sets of data from the year 2014 - 2018 specific to ports and each data set is sorted

Confusion Ma-trix Parameter	Equation
counter	$(TN+FN+FP)-TP/2TP$ (5)
Precision	$TP/TP+FP$ (6)
Recall	$TP/TP+FN$ (7)
F1	$2*(counter*recall/precision+recall)$ (8)

Table VII: Models Evaluation Matrix

and processed using the proposed machine learned methods. The machine learned methods extracted the features from randomly selected decision tree data for effective prediction from the wild. In order to evaluate the obtained results we considered the pre-defined test suites from other firewall devices with their results. This experiment setup was created on the firewall and attack strategy is applied on firewall by creating the proposed technique intelligence and while later we turned off the proposed method to record the results for evaluation. We noticed, at beginning we found few false negatives and false positives were obtained. However, these were not true in the previous vendor firewall results and from never detected by Sophos/Microsoft/Redhat and NVD database. All data sets of predicted are reevaluated using the BI faulty tolerance technique, this algorithm listed the fake/faulty data from the processed decision trees. Each data nodes from all 4 classifications are considered in this evaluation system.

5.1 PEER COMPARISON

The base testing results proved firmly the usage of lim-ited data and hence we decided to perform peer comparison test, because the peer firewall device from other vendor had sufficient vulnerability data with public recorded NVD history. The data collected from peer the firewall device is imported using the CSV and stored to SQL database. The vulnerability data analysis is made using the machine learned method and this data is classified into two categories true positive and true negative. To understand any exploit from a vulnerability and prediction found to be true then it falls into category of true positive, otherwise any exploit found to be false then we consider it as true negative. The peer comparison was conducted using the Next Generation Firewall (NGF), we applied limits on predictions on NGF available data and on proposed prediction technique. The identical test cases are created on both firewall NGF and Sophos and later while they are intermixed with the season rules. The NGF firewall produced relatively mere common scale of results with applied limits on both test cases but Sophos produced varied results on same test suite.

Year	NGF (Gaj)	Sophos	Limits Applied	+/- (Gaj)	+/- Sophos	% Diff
2014	11396	7937	1-10	17.97	11.96	6.03
2015	9781	6487	10-25	36.89	24.45	12.41
2016	9705	6447	25-50	73.04	48.52	24.52
2017	23707	14650	50-75	287.72	177.80	109.92
2018	21116	12039	75-100	370.36	211.16	159.20

Table VIII: Peer Comparison Results

Table [8] explains peer the yearly processed comparison results obtained from test suite. The NGF firewall and Sophos firewall data of 2014 to 2018 were applied to proposed machine learned methods with limits. The limits are the settings defined in the learning system for selecting the features from the decision trees. The basic trail results are not discussed here because of the absence of real-time firewall environment. The vulnerability data obtained from 2014 to 2018 is applied with the listed limits on NGF and Sophos, the test suites are repeated with combinational actions attacks on the setup. The feature selected from each firewall is listed with (+/- Gaj) and (+/- Sophos) against each year from 2014 to 2018 found to be improved. In both the test cases the prediction effi-ciency increased with real-time data from social environment and firewall history. We noticed large differences in the year 2017 and 2018 because of extreme d2web activity in the wild. The performance of proposed technique is illustrated in the Figure [8] with a detailed analysis of prediction and precision data using F1 score.

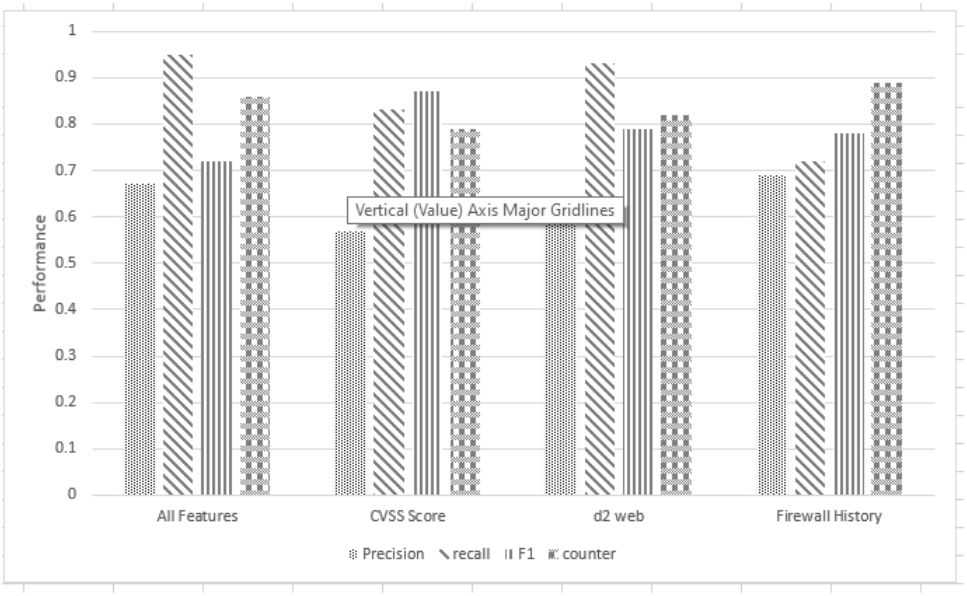

Figure 8. Prediction Results by Peer Comparision Test on All Features

In the peer comparison test, the all features and CVS scores performance dropped for F1 when applied limits on feature selection, application port, cvss score, d2web and firewall history, followed by the precision and recall. This drop of F1 signals the assessment of vulnerability data from various d2 web sources and cvss scores in real-time many times found be true negative. However, when we removed the limits on feature selection then results increased. In this setup, we focused primarily on application port attack prediction and most previous attacks and CVSS score for this specific action needs upgradation. The firewall ports and network configuration as evolved over time.

The results prove that when the test cases opened all feature selection on both the firewalls, the proposed method generated results are not commendable in comparison to application port specific prediction as shows in Figure [9]. The d2web and firewall history provides amusing prediction information about the port specific and overall. Furthermore, the vendor Sophos, Microsoft, Redhat, NGF-Gaj vulnerability data is copied from NVD data and proposed RCF based learned model is exceptional in prediction. Overall

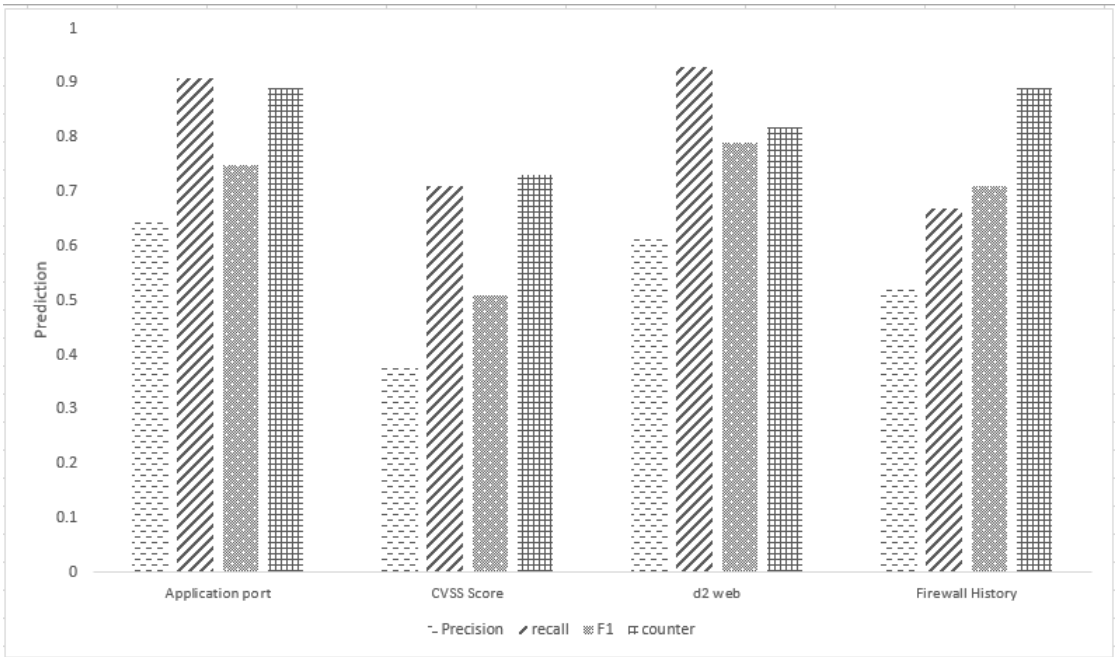

Figure 9. Prediction Results by Peer Comparision Test on Application Port

tests conducted demonstrated feasibility of cyber threat prediction. The F1 on application port specific prediction achieved 0.85 with counter 94% and recall of 0.92 on features selected.

6. ASSOCIATED WORK

To our best of learning experiences, in this paper we would be the first to showcase a unique style of countering the cyber threats by adopting behaviors as like cyber criminals and with honey trap technique. Nevertheless, sufficient cybercrime data analysis, frameworks, learning methods exist in securing the cyber world. The numerous hacking groups on d2web forums, websites and hidden tor shares are inspected in the past on cyber threat [M. Motoyama et al. 2011], [L. Allodi et al. 2017] and few investigations to discover the forthcoming risks [M. Almukaynizi et al. 2017], [N. Tavabi et al. 2018], and [J. M. Montgomery et al. 2012]. In addition, few extended and specific segment cybercriminal countering investigations [Nunes, E et al. 2016], [Robertson, J et al. 2017], [Samtani S. et al. 2016], [Soska K. et al. 2014], [Hao S. et al. 2016], [Edkrantz M. et al. 2015] but none of these investigations and proposed methods successful in discovering firewall based threats as most of the intrude operation happens through the firewall. Additionally, the Brooks-Iyengars fault-tolerance specifically predicts the faults from the machine learned data on our predictions where risks are predicted.

An outsized number of model based understanding of cyber behavior and criminal activity patterns analysis is found to propose approach to counter cyber risks. The learning based mechanism for protecting the application systems [J. Robertson et al. 2016], [M. Brown et al. 2014], the threat and counter strategy on finance application and services [A. Lendasse et al. 2000]. In this paper our research study proposes a complete real-time threat study and protection measures based on RCF mathematical modeling that is integrated to Brooks-Iyengars fault-tolerance prediction data and analysis. This work is related to live deployment and evaluated with real-world user and attackers data.

References

FIRST. LAST ACCESSED 2019. A Complete Guide to the Common Vulnerability Scoring System *https://www.cvedetails.com/cve-details.php?t=1&cve_id=CVE-2019-0708*

CAROL SABOTTKE, OCTAVIAN SUCIU,AND HSINCHNU CHEN 2015. Vulnerability disclosure in the age of social media: Exploring twitter for Predicting Real-World Exploits. *In USENIX Security Vol.15,*

LUCA ALLODI 2014. Comparing vulnerability severity and exploits using case-control studies. *ACM Transactions on Information and System Security (TISSEC) 17 Vol.1,*

BENJAMIN BULLOUGH. 2017 Predicting exploitation of disclosed software vulnerabilities using open-source data. *In Proceedings of the 2017 ACM International Workshop on Security and Privacy Analytics. ACM*

MOHAMMED ALI 2018. Analysis of Online Social Network Connections for Identification of Influential Users: Survey and Open Research Issues. *Journal ACM Computing Surveys (CSUR) Surveys Homepage archive Vol.51(1),*,Article No. 16, doi 10.1145/3155897.

MICHEL EDKRANTZ, AND ALAN 2015. Predicting Cyber Vulnerability Exploits with Machine Learning. *In SCAI* pp. 48-57

ZAKIR DURUMERIC ,JAMES KASTEN, DAVID ADRIAN,ALEX HALDERMAN,MICHAEL BAILEY, FRANK LI, NICOLAS WEAVER, JOHANNA AMANN, JETHRO BEEKMAN ,AND MATHIAS PAYER 2014 The matter of heartbleed. *In Proceedings of the 2014 Conference on Internet Measurement Conference, ACM* pp475488

KARTHIK NAYAK 2014. Some vulnerabilities are different than others. *In International Workshop on Recent Advances in Intrusion Detection. Springer* pp. 426-446

MOORE, D. 2016. Cryptopolitik and the Darknet. *Survival Vol.58(1),*pp. 7-38

RIVEST, R. L. 1978. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM Vol.21(2),*pp. 120-126

CIMPANU C. 2017. Internets Largest Bitcoinmixer. *Retrieved from https://www.bleepingcomputer.com/news/technology/internets-largest-bitcoinmixer-shuts-down-realizing-bitcoin-is-not-anonymous/*

FACHKHA, C. 2015. Darknet as a Source of Cyber Threat Intelligence: Investigating Distributed and Reflection Denial of Service Attacksr.

CIANCAGLINI, V. 2015. The Deep Web. *Trend Micro*

REELSON, A. 2016. OpenNLP Ingest Processor plugin based on Apache OpenNLP. *Retrieved from https://github.com/spinscale/elasticsearch-ingest-opennlp*

AHMIA 2014. Ahmia search engine crawler. *Retrieved from https://github.com/ahmia/ahmia-crawler*

RICHARD BROOKS 1996. Robust Distributed Computing and Sensing Algorithm. *Computer Vol.29(6),*pp. 53-60

ERIC NUNES 2016. Darknet and deepnet mining for proactive cybersecurity threat intelligence. *In Intelligence and Security Informatics (ISI), 2016 IEEE Conference on. IEEE* pp. 7-12

FIRST LAST ACCESSED 2019. Understand top trends in the threat landscape Get our perspective on 2018 cybersecurity trends such as cryptocurrency mining, supply chain attacks, and phishing in the Security. *Intelligence Report Vol.24,*

M ALMUKAYNIZI 2016. Predicting cyber threats through the dynamics of user connectivity in darkweb and deepweb forums. *In ACM Computational Social Science*

M ALMUKAYNIZI 2017. Proactive identification of exploits in the wild through vulnerability mentions online. *International Conference on Cyber Conflict (CyCon US)* pp. 82-88

FIRST LAST ACCESSED 2019. A Complete Guide to the Network Security Scan. *https://nmap.org/*

YAKKALA V. 2012. Protecting web applications from SQL injection attacks by using framework and database firewall. *Proceeding ICACCI '12 Proceedings of the International Conference on Advances in Computing, Communications and Informatics* pp. 609-613

M. MOTOYAMA 2011. An analysis of underground forums. *In Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference. ACM, 2011* pp. 71-80

L. ALLODI 2017. Economic factors of vulnerability trade and exploitation. *In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM*pp. 14831499

N. TAVABI 2018. Darkembed: Exploit prediction with neural language models. *In Proceedings of AAAI Conference on Innovative Applications of AI (IAAI2018)*

J. M. MONTGOMERY 2012. Improving predictions using ensemble bayesian model averaging. *Political Analysis Vol.20(3)*pp. 271291

ROBERTSON J. 2017. Darkweb Cyber Threat Intelligence Mining. *Cambridge University Press*

SAMTANI S. 2016. Azsecure hacker assets portal: Cyber threat intelligence and malware analysis. *In ISI* pp. 1924

SOSKA K. 2014. Automatically detecting vulnerable websites before they turn malicious. *In Usenix Security* pp. 625640

HAO S. 2016. Predator: Proactive recognition and elimination of domain abuse at time-of-registration. *In CCS2016* pp. 15681579

EDKRANTZ M. 2015. PREDICTING CYBER VULNERABILITY EXPLOITS WITH MACHINE LEARNING. *In SCAI.*

J. ROBERTSON 2016. DATA DRIVEN GAME THEORETIC CYBER THREAT MITIGATION

M. BROWN 2014. Addressing scalability and robustness in security games with multiple boundedly rational adversaries. *In International Conference on Decision and Game Theory for Security. Springer* pp. 2342

A. LENDASSE 2000. Non-linear financial time series forecasting-application to the bel 20 stock market index. *European Journal of Economic and Social Systems Vol.14(1)*pp. 8191

S. MAHANEY, AND F. SCHNEIDER 1985. Inexact Agreement: Accuracy, Precision, and Graceful Degradation. *Proc. Fourth ACM Symp. Principles of Distributed Computing* pp. 237249

Dr. Latesh Kumar obtained his Ph.D from AeU university of Malaysia, prior to this he has served in Information Technology Company Lionbridge at Mumbai, India. He later associated to Hewlett Packard, Information Technology Company, California as Service Engineer and then Technical Solution Consultant in both California and North Carolina, CO. His clients included the Data Protection, Automobile, Airports and the U.S. Army. Few years later he moved to NetApp, IT storage company and served in California and Florida as Product Partner Manger. Dr. Latesh has published numerous papers in Journals, Conference and Technical articles of IT companies like Springer, IEEE, ACM, Elsevier and IT next, NetApp Technical Library worldwide. An active Research Consultant, he has investigated several projets (Data, Security and Technology) and conducting research in cyber security and machine learning. He received several awards including distinguished Customer Excellency for Technical Commitment and Content Training at Hewlett Packard, and Best International Journal award at PSRC, Indonesia. He is professional badminton Player.

Mrs Leena H.U. is currently working as full-time research scholar in the Department of Master of Computer Applications, Siddaganga Institute of Technology, B.H. Road, Tumakuru- 572103, Karnataka, (India). She has 11+ years of experience in IT industry and moved to academics. Her research interest is in the area of netwroks, spatial analytics and Cloud computing. A grant of Rs. 3 lakh received for the research project from ICT Skill Development Society, Department of IT, BT and S & T, Govt. of Karnataka, New Age Incubation Network(NAIN) during 2016.