Botnet Attack Detection Using A Hybrid Supervised Fast-Flux Killer System

Ahmad Al-Nawasrah[1], Ammar Almomani[2,**], Huthaifa A. Al_Issa[3], Abdulellah A. Alaboudi[4], Khalid Alissa[5], Ayat Alrosan[6] and Brij B. Gupta[7,*]

[1]*Information and communication technology college, British university of Bahrain*
[2]*IT-department, Al-Huson University College, Al-Balqa Applied University, P. O. Box 50, Irbid, Jordan and Research and Innovation department, Skyline University College, Sharjah P.O. Box 1797, United Arab Emirates*
[3]*Electrical and Electronics Engineering Department, Al-Huson University College, Al Balqa Applied University, Jordan*
[4]*College of Computing and Information Technology, Shaqra University, P.O. Box 33, Riyadh, KSA*
[5]*Saudi ARAMCO Cybersecurity Chair, Department of Networks and Communication, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia*
[6]*School of Information Technology, Skyline University College, Sharjah P.O. Box 1797, United Arab Emirates*
[7]*Department of Computer Engineering, National Institute of Technology, Kurukshetra, India*
E-mail: a.alnawasrah@bub.bh; ammarnav6@bau.edu.jo; h.alissa@bau.edu.jo; alaboudi@su.edu.sa; Kaalissa@iau.edu.sa; Ayat.alrosan@skylineuniversity.ac.ae; bbgupta@nitkkr.ac.in
[*]*Corresponding Author*

Received 25 June 2021; Accepted 14 September 2021;
Publication 27 December 2021

Abstract

A Fast Flux Service Network (FFSN) domain name system method is a technique used on botnet that bot herders used to support malicious botnet

Journal of Web Engineering, Vol. 21_2, 179–202.
doi: 10.13052/jwe1540-9589.2123

actions to rapidly change the domain name IP addresses and to increase the life of malicious servers. While several methods for the detection of FFSN domains are suggested, they are still suffering from relatively low accuracy with the zero-day domain in particular. Throughout the current research, a system that's deemed new is proposed. The latter system is called (the Fast Flux Killer System) and is abbreviated as (FFKS)). It allows one to have the FF-Domains "zero-day", via a deployment built on (ADeSNN). It is a hybrid, which consists of two stages. The online phase according to the learning outcomes from the offline phase works on detecting the zero-day domains while the offline phase helps in enhancing the classification performance of the system in the online phase. This system will be compared to a previously published work that was based on a supervised detection method using the same ADeSNN algorithm to have the FFSNs domains detected, also to show better performance in detecting malicious domains. A public data set for the impacts of the hybrid ADeSNN algorithm is employed in the experiment. When hybrid ADeSNN was used over the supervised one, the experiments showed better accuracy. The detection of zero-day fast-flux domains is highly accurate (99.54%) in a mode considered as an online one.

Keywords: Hybrid supervised fast-flux, botnet detection, DeSNN.

1 Introduction

Botnets have been developing much recently. Many studies were conducted about botnets. Fast flux botnets are associated with malicious threats, like DDoS [1–4], online fraud. Whilst other systems for fast-flow detection were proposed, the detection accuracy is still low, in particular with either the zero-day domain [5–8].

The Spiking Neural Network is abbreviated as (SNN). It's a 3rd generation of the network that's neural. It seeks to have a time element added to the network [9]. Regarding eSNN, it serves as a reform made to the spiking neural network. It services as an extension for the ECOS models. It uses the firing neuron (IF). It employs Rank-Order learning which is abbreviated as (RO). The Dynamic evolving Spiking Neural Network (DeSNN) is an advancement [10, 11].

Several problems face the fast-flux Botnet detection methods. Such problems include employing the mechanism of evasion detection before having the attack launched to have the botnet malicious activities supported, such

as Denial of service attack DoS. This is faced when attempting to detect zero-day FFSNs without possessing any prior knowledge of the incoming domain name that serves the mother ship or website deemed malicious. There are problems faced in tracking the accuracy of the detection process. There are problems faced in tracking the low rates of error detection. In Ref. [12], a supervised method was used for having the fast-flux domains detected based on the ADeSNN algorithm. The proposed method introduced a hybrid method to do the same job but with improving and enhancing the performance of the classification process and gaining higher detection accuracy.

Regarding the FFKS proposed system, it seeks to deal with 2 stages to have the fast-flux domains detected. The first stage is represented in an offline learning phase that aims to have the system trained on the detection of such domains. The 2nd stage is represented in the online mode. It relies on the output reached through the first stage. Through the second stage, the online learning mode will be capable of detecting the domains called (zero-day fast-flux).

The need for such a system motivated the authors to propose the current system. The proposed system combined the classification and learning at the same time, so the system could classify the new inputs while keep learning from adding the new arrivals to the learning data. In this case, all the newly captured patterns will be included in the training set, which then helps the classifier to identify the newly arrived instances whether malicious or legitimate.

The rest of the paper is organized as follows. The related work is presented in Section 2. The proposed solution is discussed in detail in Section 3. The experiments and discussion are presented in Section 4. The conclusion has appeared in Section 5.

2 Related Work

Using the Fast flux Technique in a botnet is fully addressed and discussed in [11, 13]. In the technique suggested by Zhao and Jin [14], the classification and regression tree algorithms are employed. This system employs a dataset deemed small to identify malicious and benign FFSNs quickly. The whole system is based primarily on DNS and HTTP visiting phase and FFSN domains. Yoshioka and Matsumoto have used separate characteristics mapping [15]. It takes only just a few days for the classification task, and its detection accuracy was about 90%.

The genetic method for the detection of the problem of fast-flux domains in real-time was developed by Lin, et al. [16]. The procedure proposed for having the benign and the fluxed domains classified through using a two detection structure. First the domain name entropy (E-DPN) of the flux-agent node preceding this node. Lastly, the default Round Trip Time Deviation (SD-RTT) between all the flux-agent return IPs and the user. The whole geographic characteristic is used to calculate the number of various IPs & ASNs in the DNS answer. Unfortunately, The botmaster evaded these two detection features [17, 18]. The genetic algorithms show a good accuracy level (as's mentioned in this research). However, their results could have been affected by the return of a list consisting of several IP addresses which belong to the exact AS. Based on the previous implementations, the classifier shows an overall accuracy of (95.37%). The function of the linear decision, which is employed as the classification process must have the categorizer of the linear function estimated. If not, the process of classification shall involve errors deemed significant [19].

The genetic method to the detection of the problem of the fast-flux domains in real-time was addressed by Ref. (2013).

Instead, the C4.5 algorithm [20] was presented at the event. Numerous feature sets have been evaluated to have a fast-flux network detected. The feature sets include timing, domain-based, network-specific, and DNS feature sets. The data size is deemed small. The level of accuracy of the concerned experiment is deemed high. Furthermore, where all of the features of the experiment are engaged, the results of the prediction are considered insensitive to the two main features (domain and time feature sets) [18]. Moreover, since C4.5 is a supervised learning algorithm, the undiscovered attack, particularly the zero-day fast-flux domain, could not be discovered. Furthermore, the accuracy wasn't quite as high as in their work, according to our implementation; it was 93.38%.

The system called (fast-flux botnet detection) was proposed in [12], authors used the ADeSNN algorithm for the detection of fast-flux domains in a mode deemed supervised. Their results were promising. However, as their classifier worked in supervised learning more, that leads to disability to detect fast-flux zero-day domains. The best accuracy that they gained was 98%.

In [21], L. Yang and G. Gan proposed a new feature named the ratio of the name of the authoritative server in the traffic and used the random forest classifier to judge their proposed feature. They stated that their proposed system is light and accurate. However, they relayed passive traffic data via

Wireshark therefore they work offline. Also, the highest gained accuracy reached 98%.

In [22], Random Forest (RF) classifier was applied on ten selected features, seven of them newly used. According to their experiments, the accuracy was 95%. But the proposed method was based on passive learning mode.

Based on that, the need for a hybrid system that detects new instances online and learns from the new incoming cases to enhance the classification process for the lifelong becomes mandatory, which is similar to self-supervised learning [23]. The rest of this proposed work will present a kind of what the authors are going to propose according to this issue.

3 Hybrid Fast Flux Killer System

In the present part, further details shall be presented about the FFKS. That includes supervised online and offline phases. The researcher sheds a light on FFKS's life cycle. That is done by beginning from the basis of the system that is at the offline phase till shedding a light on the process of having the zero-day fast-flux domains detected at the phase called the online phase.

3.1 The FFKS offline Phase

The offline phase seeks to deal with data that are labeled. It seeks to deal with this data before working with the online phase discussed in Ref. [12]. While the operation of learning is progressing, the classification threshold shall be improved and developed. When the process of learning reaches the end, the output of the latter phase shall serve as the final weights of the output spiking neurons and the classification threshold. Regarding the weights, they're saved in the weights repository. Also, the threshold shall get stored.

3.2 The FFKS online Phase

In the present section, the online phase shall get introduced as an element of the system abbreviated as (FFKS). Regarding that phase, it seeks to deal with the domains. Thus, the online learning element of the system shall be capable of dealing with data deemed as unknown. In this case, the data is the zero-day fast-flux domains. Regarding the process of classification, it is based on the output of the previous offline phase.

The novelty of this work reveals a new era of evolving the newly arriving data into the learning process, while the system can do the classification

process. Based on this, any other solution could adopt the idea of combining learning and classification into the same system which helps to enhance the performance of the system to be a multi-purpose platform to solve a different kinds of issues and problems.

The following discussion presents the stages of the system platform.

3.2.1 The pre-processing stage

The fast-flux public dataset was found as stacks of DNS responses; a script of python was written for having the required information extracted to create the set of features. Some features need contacting the ASN to obtain additional information on the IP addresses. A local ASN repository is allocated to accelerate the operation of creating the feature set. This repository is located in the local dive of the concerned system. That can be seen in Figure 1.

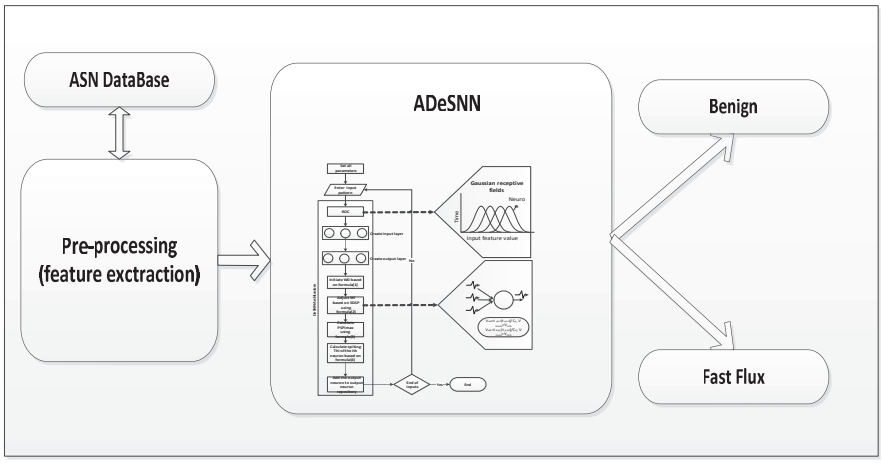

Figure 1 The pre-processing phase.

3.2.2 Feature extraction

The present stage aim at having some features created and calculated. As it's shown in Ref. [12], several features must be calculated based on the additional information that's offered by a third-party database. Several features can be obtained directly from the DNS response message.

3.3 The system called (Hybrid Fast Flux Killer System)

Both stages were joined with each other to meet the intended research goal. Such a goal is represented in having the zero-day fast-flux domains detected in a mode deemed online.

Regarding the system of FFKS, it begins with the stage of supervision to develop the main seeds of the operation of classification. After that, the mode of the online detection shall begin to have the zero-day fast-flux domains detected. When this phase ends, the offline learning mode shall be re-executed another time to have criteria of classifications refined.

Based on the second figure, the phase deemed offline shall receive the inputs that are labeled one by one. It shall create the SNN in a manner that is based on (ADeSNN). While having further inputs arriving, the SNN shall become bigger. It's learning from the inputs shall generate the matrix of the final weights. Afterward, for every input record, an output neuron shall be developed for creating the input pattern along with the operation of learning. At the last part of this stage, the whole output neurons' weights were saved in the weight repository. During the operation of learning, the classification threshold got computed to have it employed in the operation of classification. That's done following measured related to similarity.

Regarding the online mode, it seeks to deal with the testing data. As for the ADeSNN algorithm, it shall capture the domains' features. After that, it shall have the ADeSNN trained about the inputs considered new. It shall access the classification threshold which is saved at the offline phase for classifying the unknown domains. While the records deemed new get trained by the ADeSNN in a mode considered online, the final weights shall become ready to be saved in the weights repository.

Regarding the new weights of the input records that are new which are saved at the weights repository, they shall be utilized after having a specific number of records and at a specific time. In the present case, they shall be utilized after having one thousand records, in a learning mode deemed offline. That shall be done to improve the value of the classification threshold as the new inputs. They shall serve as an element of the training dataset of the phase deemed offline. Regarding the latter process, it can be shown in the second Figure 2.

Regarding the classification threshold and the weight repository, they facilitate the process of storing on the memory. There isn't any need for having all the incoming inputs stored forever. It is just enough to have a

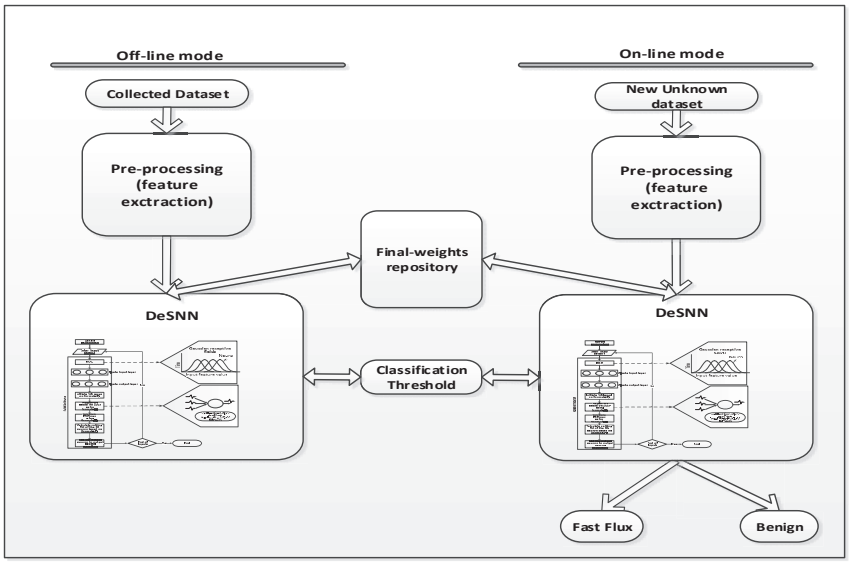

Figure 2 The hybrid FFKS.

specific number of needed inputs stored. In this case, it's enough to store one thousand records.

3.3.1 Dataset

The hybrid FFKS proposed method must be assessed and tested in terms of the ability to have fast-flux domains detected. Thus, the researcher used the exact public dataset that was employed in Ref. [12]. This dataset was employed in ref. [24]. Most of the systems to learning machine systems employ the same sources [13, 25–31]. Such a dataset is composed of DNS answers labeled as legitimate and fast-flux domain names. As for the domains deemed legitimate, they're are selected from Alexa, a leading blogger as Blogs on top "BOT," among the most trusted websites. In contrast, fast-flux domains from well-known known blacklisted fast-flux websites [6]. For identifying every class, the chosen feature set is included in each record. Instances (1710) are presented in fast-flux data sets, and legitimate data sets (3420).

3.3.2 Feature set

Feature extraction is the first step of the proposed solution. The well-built method of fast-flux detection for botnet should differentiate between a network deemed legal and a network deemed malicious. However, a well-built Fast Flux Network (FFN), because DNS records belonging to the same geographic areas, appear to be a benign CDN. It makes the detection system have such kinds of FFN domains misclassified as being benign CDN depending on the IP address feature. Furthermore, although the amendment influences the performance level of FFN, the developers of FFN seek to make changes to the features of the fast-flux network for evading detection. A new method of detection therefore should depend on the features of the FFN, as they are not susceptible to rapidly adjust.

Considering the existent fast-flux dataset, several features shown in the current system have already been used [12]. The process of classifying legitimate domains and fast-flux can be strengthened and improved by new features. The features that were chosen are shown in the table below

Table 1 The proposed set of features

Feature	Description
IPans	Number of IP addresses in the answer section
NSadd	Number of IP addresses in the additional section
NASN_ans	Number of ASN for the IP addresses of the answer section
NASN_add	Number of ASN for the IP addresses of the answer section
AVGSIM	The average of similarity of the ASN (among the answer section and the ASN of the victim himself)
Qtime	Time of the query
Msgs	Message size

Through this Table 1, a description of all the features in the set is shown. The first couple of features are straightforward and obtained from the response directly. The second and third features need to get further information from the ASN local repository. The fifth feature seeks to compute the average of the similarity that exists between the ASN number of the IP address that's requested from one hand and the other ASN number of the IP addresses that's returned in the response of DNS. It's defined as the average similarity in the response section of the response of DNS between user IP and proxy bots, which is calculated by Equation (1) [32]:

$$\mathrm{M_i}y(e) - x(e)) = 1 - \frac{\sum_{j=1}^{n} yij(e) - xij(e)}{\sum_{j=1}^{n} yij(e) - xij(e)} \tag{1}$$

ė stands for the input vector. $\mu(\dot{e})$ & $\delta(\dot{e})$ are deemed much similar in case M(μ(ė),δ(ė)) $\geq$ 1/2. This means that the greater the value is, the most similar the variables are.

4 Discussion & Experiment

Online and offline learning phases operate simultaneously during the first step. The model deemed offline shall have an algorithm trained on the fast-flux data. Following that, it shall have the outcomes of the process of classification produced and delivered to the following step. In the 2nd step, the phase of online learning shall be dealing with new input records. It shall have the algorithm used for the production of the weights that are the final ones. Following that, it shall employ the outcomes reached through step 1 to assist the online phase in the process of classifying the new-arriving inputs in case inputs are fast-flux domains.

The software and hardware that are used in evaluations were built on the current Linux mint operating system: i7 core 7500U CPU, 16 GB RAM, and comparative method simulations were carried out through employing MATLAB 8.5 in addition to Python 2.7 environments.

For the assessing system called (the Hybrid FFKS), the public fast-flux data was employed. It was used for making a comparison between the reached results and the ones reached through the online supervised stage in the other paper. 3-folded cross-validation is carried out. 3 trials were carried out for meeting such a goal. Every trial in the 1st couple of folds is carried out in the offline learning mode to have the ADeSNN trained. The 3rd fold was fed into the mode called (online learning). The architecture of the ADESNN is presented in the following figure.

At the phase deemed offline, the first couple of folds is employed to have the algorithm called (ADeSNN) trained on the benign and fast-flux domains. The initial weights of SNN get updated following the new inputs. The final weights shall get saved at the weights repository. Regarding the classification threshold, they shall be trained for having the benign and the fast-flux domains classified. The final value of the saved threshold shall be employed by the following step at the phase deemed online.

As for the third fold, it shall be fed to the phase deemed online as being new data inputs. As for the algorithm called ADeSNN, it shall implement the SNN have the final weights of such inputs produced. The phase deemed online can have the classification threshold -produced through the phase deemed offline- accessed. The algorithm called ADeSNN shall be employed

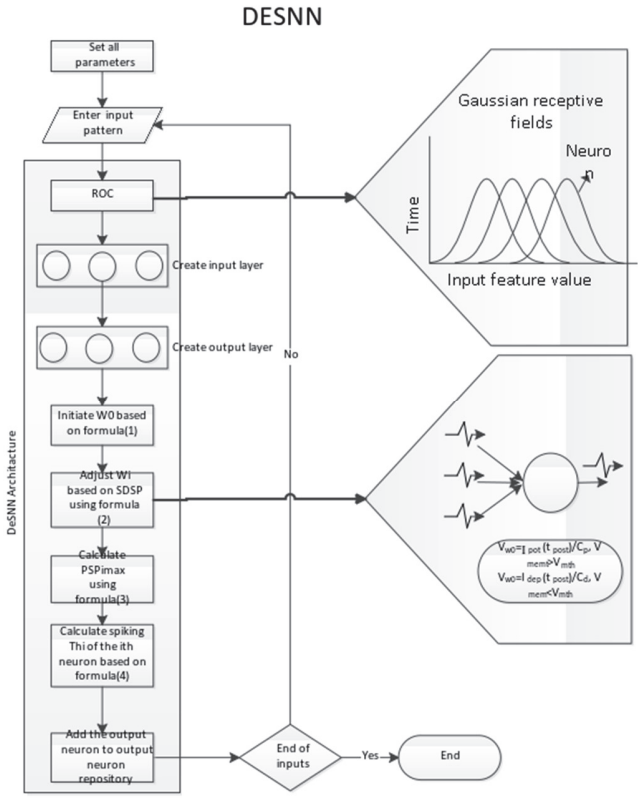

Figure 3 ADeSNN architecture.

for having the new inputs classified. Following that, the new weights shall get added to the weight repository or replaced.

For each one thousand input records that are new, the system shall have the offline phase re-trained on the weights that are saved in the weights repository. That shall be done to improve the classification threshold. Thus, the system that's proposed shall seek to update its classification ability following the changes that occur to the input. That shall grant the system workability that is characterized as being a lifelong one. The functionality of the system adapts with the recent changes that occur to the form

4.1 Experiments and Results

Based on the 3 main experiments, the system called (hybrid FFKS) operated effectively in the mode deemed offline and the one deemed online. It operated effectively in a model deemed hybrid for having fast-flux domains detected in a mode deemed offline. It operated effectively for having the algorithm trained on the domains of (zero-day fast-flux) in a mode deemed online. As for the average of the experiments' results, they are shown in the second table.

A group of accuracy measures was conducted to evaluate the results, such as False Negative Rate FNR, False Positive Rate FPR, True Positive Rate TPR, True Negative Rate TNR, Accuracy ACC, Precision, Recall, F-measure, The Matthews Correlation Coefficient MCC, Root mean square error RMSE, Non-Dimensional Error Index NDEI. The AUC represents the performance level shown by the classifier [33]. Likewise, the AUC serves as a robust estimator for the performance level shown by the classifier [34].

Table 2 The results that are reached through the hybrid FFKS

Evaluation Measures	Hybrid FFKA
FNR	0.00%
FPR	0.059%
TPR	100.00%
TNR	99.41%
ACC	99.54%
Precision	99.41%
Recall	100.00%
F-measure	99.71%

Based on the Table 2, the system is capable of having benign domains classified with a TPR of 100%. It is capable of having the fast-flux domains classified with a showing FPR of 0.59%. Every measure for measuring the accuracy of the system is listed in the second table. The error results are shown in the Figure 7.

Figure 4 displayed the results which are reached through the system called (hybrid FFKS). Based on the error estimators, the latter system is capable of minimizing the number of instances that were misclassified. It's capable of doing that based on the new enhancements.

It's proved that the latter system is capable of having the domains called zero-day fast-flux detected in a mode deemed online. The total accuracy shown by the system in this regard is 99.54%. The system improves its

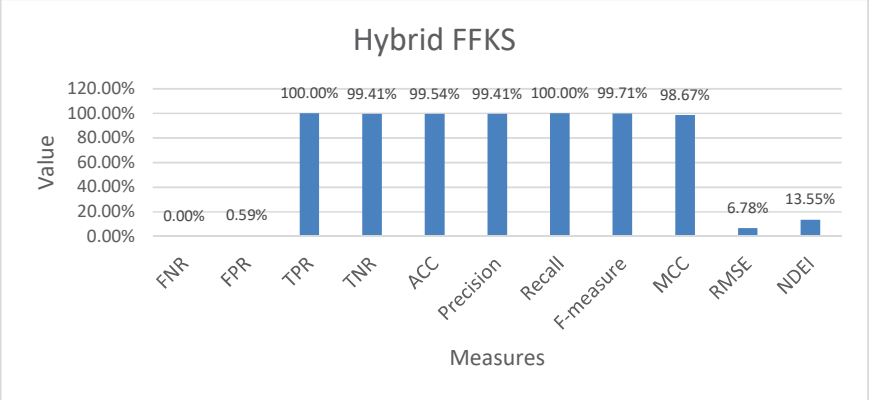

Figure 4 The general idea of combining multiple models.

accuracy periodically in the process of classification in the mode deemed offline.

4.2 Comparison Between the Online Supervised System and the Hybrid Ones

As for the supervised system, the hybrid system is introduced to have the system trained on the process of having the fast-flux domain detected. It created the classification threshold of the online phase and improved it.

The argument below sheds a light on the comparison made between the supervised system phase from the fast-flux botnet catcher system [12] (i.e. the supervised FFKS) and the FFKS hybrid offline and online system. The researcher used a dataset that's public to have the selected classifiers tested. 3 trials were made through the use of a public fast-flux dataset. They aim to assess the capability of the FFKS system in having the problem of the fast-flux domains solved. The results reached through the experiments and the performance level are deemed promising. They indicate that there is a rise in the detection-based accuracy level of fast-flux domains. To have the quality level of the learning phase of the ADeSNN measured, a 3-folded cross-validation technique is employed for having the algorithm's error rate estimated. The three trials were carried out. The average was calculated then. The third table offers a summary for the comparison of the results that are obtained in these trials and the trials in [12].

Table 3 shows the results of the comparison made between the supervised phase experiment fast-flux botnet catcher system in [12] and the hybrid FFKS

Table 3 Results of making a comparison between the FFKS supervised system in[12] and the FFKS hybrid system

Evaluation Measures	Supervised FFKA [8]	Hybrid FFKA
FNR	0.00%	0.00%
FPR	2.41%	0.59%
TPR	100.00%	100.00%
TNR	97.59%	99.41%
ACC	98.76%	99.54%
Precision	97.53%	99.41%
Recall	100.00%	100.00%
F-measure	98.75%	99.71%
MCC	97.56%	98.67%
RMSE	11.11%	6.78%
NDEI	22.19%	13.55%

experiment. Through having such results analyzed, the two systems show similar achievement in terms of having the benign domains detected. As for the detection rate of the benign domains, it's represented in one hundred terms. In terms of having the fast-flux domains detected, the system deemed hybrid shows a better performance. It shows a detection accuracy rate of 99.54%. As for the supervised online system, it shows a percentage of 98.76% in this regard [12]. As shown in Figure 5.

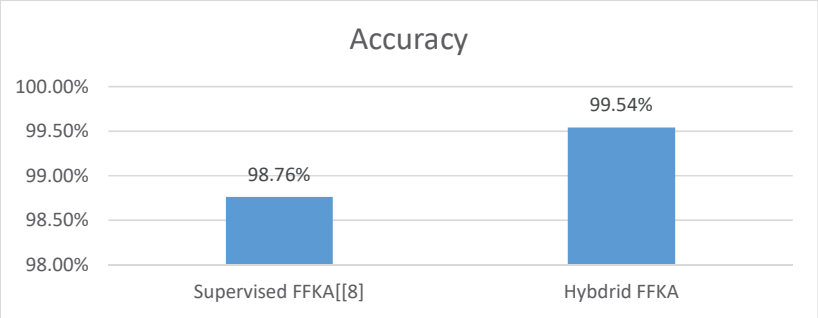

Figure 5 An accuracy comparison was made between the online supervised system and the hybrid FFKS one.

The proposed system proved better accuracy over the supervised system other measures were used. They are displayed in the third table. They include Precision and MCC. They include F-measure and recall. They all indicate that the hybrid system shows a better performance level than the online one.

Figure 6 presents the compared graph that presents the whole measures in the online supervised experiment and the hybrid FFKS system experiment.

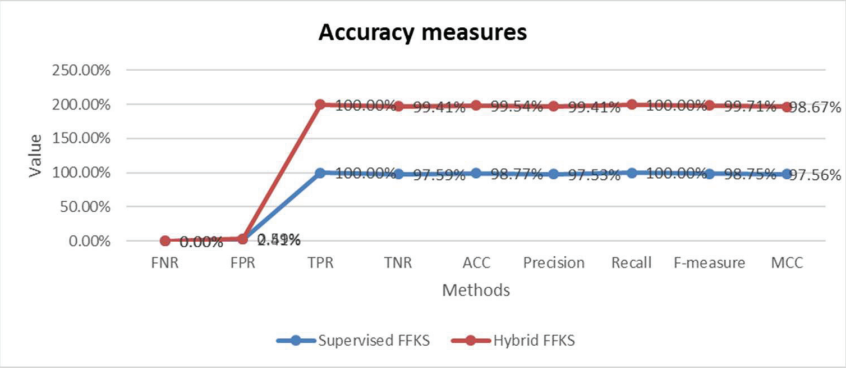

Figure 6 A comparison made between the online supervised system and hybrid FFKS one.

RMSE and NDEI of error estimator show a rise in comparison to the same in the supervised phase. Thus, the percentages of the misclassification of the normal and fast-flux domains were reduced in the system considered hybrid. That can be seen in Figure 7.

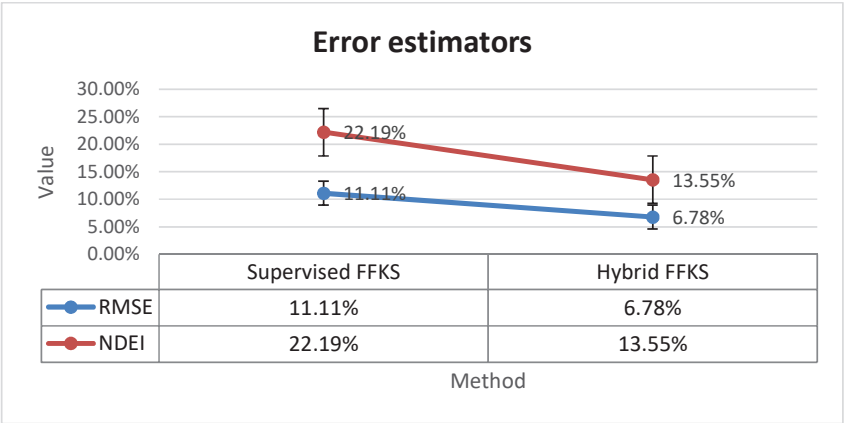

Figure 7 A comparison made between the supervised system and the hybrid FFKS one in terms of error.

The contribution made by the FFKS enhanced the accuracy level and the capability to have fast-flux domains in general and zero-day fast-flux domains in particular detected in a mode deemed hybrid. Based on f-measure

and precision, there is an improvement in their values in comparison to the previous trial results. The researcher found that the detection accuracy of the FFKSA is a hybrid system that dramatically enhanced the capability of the algorithm called ADeSNN in having the incoming inputs classified.

5 Limitations and Future Work

There is a shortcoming in the DeSNN algorithm. It's represented in the fact that a great number of parameters must be set before running them. Through this research, the researcher addressed a problem. Still, there is a need for setting numerous parameters. The problem of fast flux hasn't been solved yet. Thus, more effort should be exerted in this regard by the private sector and the public one. There must be a coronation between public and private organizations in this regard. That must be done to make sure that the dataset is tested and controlled correctly.

Researchers should conduct studies about the use of fuzzy rules rather than weight matrices.

6 Conclusion

Botnets have been developing much. Many studies have been conducted about them. Fast flux botnets are associated with malicious threats, like DDoS, online fraud. Whilst other systems for fast-flow detection were proposed, the detection accuracy is still low, in particular with either the zero-day domain

The researcher provided a system for having the fast-flux domains detected and classified. This system is called (Fast Flux Killer). It's developed through the use of ADeSNN. It's capable of having FF-domains detected in a mode deemed hybrid. It can do that during the zéro days in online mode. It proved its effectiveness in having fast-flux domains detected with showing a high level of accuracy.

An evaluation has been made with a previous supervised version of the detection system. This comparison indicates that the system being proposed outperforms the other system. A public dataset used in trials to demonstrate the impacts of the ADeSNN hybrid algorithm, high accuracy achieved in the detection of fast-flux domains is (99.54%) approximately in a mode deemed online and improved by offline mode.

References

[1] G. I. Shidaganti, A. S. Inamdar, S. V. Rai, A. M. J. I. J. o. C. A. Rajeev, and Computing, "Scef: A model for prevention of ddos attacks from the cloud," vol. 10, no. 3, pp. 67–80, 2020.

[2] A. Dahiya and B. B. J. F. G. C. S. Gupta, "A reputation score policy and Bayesian game theory based incentivized mechanism for DDoS attacks mitigation and cyber defense," vol. 117, pp. 193–204, 2021.

[3] K. Bhushan, B. B. J. J. o. A. I. Gupta, and H. Computing, "Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment," vol. 10, no. 5, pp. 1985–1997, 2019.

[4] M. Chhabra, B. Gupta, and A. Almomani, "A novel solution to handle DDOS attack in MANET," 2013.

[5] K. Alieyan, A. Almomani, M. Anbar, M. Alauthman, R. Abdullah, and B. Gupta, "DNS rule-based schema to botnet detection," *Enterprise Information Systems*, pp. 1–20, 2019.

[6] K. Alieyan, M. Anbar, A. Almomani, R. Abdullah, and M. Alauthman, "Botnets Detecting Attack Based on DNS Features," in *2018 International Arab Conference on Information Technology (ACIT)*, 2018, pp. 1–4: IEEE.

[7] K. Alieyan, A. Almomani, R. Abdullah, and M. Anbar, "A Rule-based System to Detect Botnets based on DNS," in *2018 8th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2018, pp. 115–120: IEEE.

[8] A. Almomani, O. M. Dorgham, M. Alauthman, M. Al-Refai, and N. Aslam, "Botnet Behavior and Detection Techniques: A Review," *Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives*, p. 223, 2018.

[9] A. Almomani, M. Alauthman, M. Alweshah, O. Dorgham, and F. Albalas, "A comparative study on spiking neural network encoding schema: implemented with cloud computing," *Cluster Computing*, vol. 22, no. 2, pp. 419–433, 2019.

[10] N. Kasabov, K. Dhoble, N. Nuntalid, and G. Indiveri, "Dynamic evolving spiking neural networks for on-line spatio-and spectro-temporal pattern recognition," *Neural Networks*, vol. 41, pp. 188–201, 2013.

[11] A. Al-Nawasrah, A. A. Almomani, S. Atawneh, M. J. I. J. o. C. A. Alauthman, and Computing, "A Survey of Fast Flux Botnet Detection With Fast Flux Cloud Computing," vol. 10, no. 3, pp. 17–53, 2020.

[12] A. A.-N. Ammar Almomani, Mohammad Alauthman,Mohammed Azmi Al-Betar, Farid Meziane., "Botnet Detection Used Fast-Flux Technique, Based on Adaptive Dynamic Evolving Spiking Neural Network Algorithm," ed. Journal International Journal of Ad Hoc and Ubiquitous Computing, 2020.

[13] A. Al-Nawasrah, A. Al-Momani, F. Meziane, and M. Alauthman, "Fast flux botnet detection framework using adaptive dynamic evolving spiking neural network algorithm," in *2018 9th International Conference on Information and Communication Systems (ICICS)*, 2018, pp. 7–11: IEEE.

[14] Y. Zhao and Z. Jin, "Quickly Identifying FFSN Domain and CDN Domain with Little Dataset," 2015.

[15] Y. M. P. Pa, K. Yoshioka, and T. Matsumoto, "Detecting malicious domains and authoritative name servers based on their distinct mappings to IP addresses," *Journal of information processing*, vol. 23, no. 5, pp. 623–632, 2015.

[16] H.-T. Lin, Y.-Y. Lin, and J.-W. Chiang, "Genetic-based real-time fast-flux service networks detection," *Computer Networks*, vol. 57, no. 2, pp. 501–513, 2/4/ 2013.

[17] F.-H. Hsu, C.-S. Wang, C.-H. Hsu, C.-K. Tso, L.-H. Chen, and S.-H. Lin, "Detect fast-flux domains through response time differences," *Selected Areas in Communications, IEEE Journal on*, vol. 32, no. 10, pp. 1947–1956, 2014.

[18] T. Otgonbold, "ADAPT: An anonymous, distributed, and active probing-based technique for detecting malicious fast-flux domains," 2014.

[19] P. S. Chahal and S. S. Khurana, "TempR: Application of Stricture Dependent Intelligent Classifier for Fast Flux Domain Detection," *International Journal of Computer Network & Information Security*, vol. 8, no. 10, 2016.

[20] Z. B. Celik and S. Oktug, "Detection of fast-flux networks using various dns feature sets," in *Computers and Communications (ISCC), 2013 IEEE Symposium on*, 2013, pp. 000868–000873: IEEE.

[21] L. Yang and G. Gan, "Research and Detection of Fast-flux Botnet," in *IOP Conference Series: Earth and Environmental Science*, 2021, vol. 693, no. 1, p. 012031: IOP Publishing.

[22] D.-T. Truong, D.-T. Tran, and B. J. J. o. I. T. Huynh, "Detecting Malicious Fast-Flux Domains Using Feature-based Classification Techniques," vol. 21, no. 4, pp. 1061–1072, 2020.

[23] D. Yuan, X. Chang, P.-Y. Huang, Q. Liu, and Z. J. I. T. o. I. P. He, "Self-supervised deep correlation tracking," vol. 30, pp. 976–985, 2020.

[24] S.-Y. Huang, C.-H. Mao, and H.-M. Lee, "Fast-flux service network detection based on spatial snapshot mechanism for delay-free detection," presented at the Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, Beijing, China, 2010.

[25] T. Holz, C. Gorecki, K. Rieck, and F. C. Freiling, "Measuring and Detecting Fast-Flux Service Networks," in *NDSS*, 2008.

[26] Y. Sheng, Z. Shijie, and W. Sha, "Fast-flux attack network identification based on agent lifespan," in *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*, 2010, pp. 658–662.

[27] B. Yu, L. Smith, and M. Threefoot, "Semi-supervised Time Series Modeling for Real-Time Flux Domain Detection on Passive DNS Traffic," in *Machine Learning and Data Mining in Pattern Recognition*, vol. 8556, P. Perner, Ed. (Lecture Notes in Computer Science: Springer International Publishing, 2014, pp. 258–271.

[28] S. Martinez-Bea, S. Castillo-Perez, and J. Garcia-Alfaro, "Real-time malicious fast-flux detection using DNS and bot related features," in *PST*, 2013, pp. 369–372.

[29] M. T. Qassrawi and H. L. Zhang, "Detecting Malicious Fast Flux Domains," in *Applied Mechanics and Materials*, 2012, vol. 157, pp. 1264–1273: Trans Tech Publ.

[30] C.-M. Chen, S.-T. Cheng, and J.-H. Chou, "Detection of Fast-Flux Domains," *Journal of Advances in Computer Networks*, vol. 1, no. 2, 2013.

[31] C. Castelluccia, M. A. Kaafar, P. Manils, and D. Perito, "Geolocalization of proxied services and its application to fast-flux hidden servers," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, 2009, pp. 184–189: ACM.

[32] S. Alkhazaleh, A. R. Salleh, and N. Hassan, "Possibility fuzzy soft set," *Advances in Decision Sciences*, vol. 2011, 2011.

[33] J. A. Swets, *Signal detection theory and ROC analysis in psychology and diagnostics: Collected papers*. Psychology Press, 2014.

[34] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 6// 2006.

Biographies

Ahmad Al Nawasrah (a.alnawasreh@bub.bh) received his Ph.D. in Computer Science-Information security from the University of Salford, the UK in 2018. Currently, Dr. AlNawasrah is an assistant professor at ICT college, British University of Bahrain. He has published several research papers in International Journals and Conferences with a high reputation, where some of these publications are tracked by Thomson Reuters (ISI) and Scopus. His research interests lie in Information Security, Internet cyber-crimes.

Ammar Almomani received his Ph.D. Degree from Universiti Sains Malaysia (USM) in 2013. He has published more than 75 research papers in International Journals and Conferences of high repute including IEEE, Elsevier, ACM, Springer, Inderscience, etc. with many international awards, He has visited several countries to present his research work, he is serving as a reviewer for 10s Journals of IEEE, Springer, Wiley, Taylor & Francis, etc. he has 16 years of experience with taught more than 40 different subjects in computer science, networks, and cybersecurity, and programming language, he has many international certificates and participation in dozens of projects and specialized scientific courses, His research interest includes cybersecurity, advanced Internet security, and monitoring, he is an associate professor and senior lecturer at Al- Balqa Applied University and currently he is a professor and ahead of research and innovation department in SKYLINE

university college-SHARJAH-UAE. link: https://scholar.google.com/citations?user=d_tRtPkAAAAJ&hl=en

Huthaifa A. Al_Issa received his Bachelor's and Master's degrees in Electrical and Computer Engineering at the Near East University, Cyprus, in 2003 and 2005, respectively with high honors GPA. He received his Ph.D. degree in Electrical Engineering at the University of Dayton, Dayton, OH, USA, in 2012. Currently, he is an assistant professor in the Department of Electrical and Electronics Engineering at AL Balqa Applied University, Al-Huson University College. He has been a member of the Jordan Engineers Association (JEA) since 2003.

Abdulellah A. Alaboudi received a master's degree and a Ph.D. degree in computer sciences from the University of Staffordshire, U.K. He is currently working at Shaqra University, Saudi Arabia, as an Assistant Professor. He has vast experience as a Business Process Reengineer and project management. An ample number of peer-reviewed articles are on his credit. His research interests include the Internet of Things, cybersecurity, software engineering, wireless networks, and machine learning.

Khalid Alissa, received a Ph.D. degree in Information security (Access control) from Queensland University of Technology Brisbane in 2010–2015. He is assistant professor at college of computer sciences and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia, and National Center for Satellite Technologies, King Abdulaziz City for Science and Technology (KACST), Riyadh, Saudi Arabia. His research interest includes information security.

Ayat Alrosan received a Ph.D. degree from Universiti Sains Islam Malaysia (USIM) in 2017. She has published many research papers in International Journals and Conferences of high repute. Currently, she is an assistant professor at Deanship of Information and Communication Technology, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. His research interest includes image processing, data clustering, and optimization, aaalrosan@iau.edu.sa, https://orcid.org/0000-0001-9400-4077. AYAT's photograph is not available at the time of publication.

Brij B. Gupta received PhD degree from Indian Institute of Technology Roorkee, India in the area of information security. He has published more than 300 research papers in international journals and conferences of high repute. He has visited several countries to present his research work. His biography has published in the Marquis Who's Who in the World, 2012. At present, he is working as an Assistant Professor in the Department of Computer Engineering, National Institute of Technology Kurukshetra, India. His research interest includes information security, cyber security, cloud computing, web security, intrusion detection, computer networks and phishing.