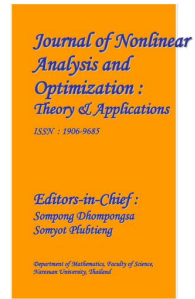


Journal of Nonlinear Analysis and Optimization

Vol. 15, Issue. 1, No.15 : 2024

ISSN : **1906-9685**



CYBER ATTACKS DETECTION USING MACHINE LEARNING

Thadvai Relax(20641A6761), Btech Student, CSE (Data Science), Vaagdevi College of Engineering

Ponnaveni Mahesh(20641A6754), Btech Student, CSE (Data Science), Vaagdevi College of Engineering

Mohammed Asman Zameeruddin (20641A6743), Btech Student, CSE (Data Science), Vaagdevi College of Engineering

Vemula Sai Kumar (20641A6764), Btech Student, CSE (Data Science), Vaagdevi College of Engineering

Mrs. Zareena Begum, Assistant Professor, CSE (Data Science), Vaagdevi College of Engineering

ABSTRACT

The increased usage of cloud services, growing number of web applications users, changes in network infrastructure that connects devices running mobile operating systems and constantly evolving network technology cause novel challenges for cyber security. As a result, to counter arising threats, network security mechanisms, sensors and protection schemes also have to evolve, to address the needs and problems of the users. In this article, we focus on countering emerging application layer cyber-attacks since those are listed as top threats and the main challenge for network and cyber security. The major contribution of the article is the proposition of machine learning approach to model normal behavior of application and to detect cyber-attacks. The model consists of patterns (in form of Perl Compatible Regular Expressions (PCRE) regular expressions) that are obtained using graph-based segmentation technique and dynamic programming. The model is based on information obtained from HTTP requests generated by

client to a web server. We have evaluated our method on CSIC 2010 HTTP Dataset achieving satisfactory results

1. INTRODUCTION

Recently the number of security incidents reported all over the world has increased. The national CERTs (e.g. CERT Poland [1]) report that the number of attacks has increased significantly when compared to the previous years. According to the report [1] in 2012 there were 1082 incidents, which is an increase of nearly 80% in comparison to the previous year, mainly due to malware and phishing. The increased number of incidents is strongly related to the increased number of mobile device users who form the population of connect-from-anywhere terminals and regularly test the traditional boundaries of the network security. Also, the so-called BYOD (bring your own device [4]) trend exposes the traditional security of many enterprises to novel and emerging threats. Many of nowadays malwares like ZITMO (Zeus In The Mobile) do not aim at mobile device itself but rather on gathering the information about the users, their private data and gaining the access to remote services like banks and web services. There is also a significant number of reported incidents that relate to a huge widespread adoption of the social media. This trend has an impact on accelerated spread of different kinds of malwares and viruses. As reported by Sophos Labs [2] in 2013, botnets have become more widespread, resilient and camouflaged and they are finding some dangerous new targets. Moreover, since cloud services and SaaS have been adapted by small and medium enterprises, a big challenge for network security arises. Such companies store, maintain and transport crucial data using third party infrastructure where traditional points of inspection cannot be deployed. This trend is connected with the criminals that see the potential to get more return on their investment with cloud attacks, since they only need to ‘hack one to hack them all’. Other well-known problems like attacks on the web applications to extract data or to distribute malicious code remain unsolved. Cybercriminals continuously steal data and distribute their malicious code via legitimate web servers they have compromised. Moreover, as it is shown in the Figure 1, the attacks on web applications constitute more than a half of all incidents identified by Kaspersky Lab [13]. The list

of top 10 most critical risks related to web applications security, provided by OWASP (Open Web Application Security Project) indicates ‘Injection’ (including Structured Query Language (SQL), Operating System (OS) and Lightweight Directory Access Protocol (LDAP) injections) as a major vulnerability [5]. Factors, such as easy exploitability and severe impact of potential attacks are mentioned as the most crucial. To perform an injection attack, the attacker sends a simple text that exploits the syntax of the targeted interpreter, and therefore almost any source of data can be an injection attack vector. A successful injection can cause serious consequences including data loss, corruption, lack of accountability or the denial of access. Additionally, the level of prevalence is described as common, while level of detectability is identified as average [5]. Therefore, in this article we focus on detecting emerging application layer attacks. The major contribution of this article is the proposition of a machine learning technique to model normal behaviour of application and detect cyber attacks. The article is structured as follows: in Section 2, the overview of cyber attack detection techniques based on machine learning is provided. In Section 3, the proposed method is described in detail. In Section 4, the benchmark database used in our experiments is discussed. The experimental set-up and results are presented in Section 5. Conclusions are given thereafter.

2.LITERATURE SURVEY

MLBSA This section proposes an attack methodology for stealing controlled information attacks utilizing ML techniques, And the methodology is named as the Machine Learning Based Stealing Attack (MLBSA) [19] methodology. We revised the cyber kill chain for modelling the MLBSA methodology. A typical kill chain consists of seven stages including reconnaissance, weaponization, delivery, exploitation, installation, command and control, and actions on objectives. Reconnaissance aims to identify the target by assessing the environment. As a result, the prior knowledge of attacks [20] can guide data collection. Regarding the ML-based stealing attack, weaponization means data collection. Extracting the useful information via feature engineering is essential. Using supervised learning, the ML-based [11] model is built as a weapon taking actions on objectives. Moreover, the ML-based stealing attack may keep improving its performance and accumulate the knowledge gained from its retrieved results. Other stages of kill chain, including delivering the weapon to the victim, exploiting the vulnerabilities, installing the malware, and using command channels for remote control, are considered as a preparation phase before attacking the objectives. In this paper, the preparation phase is named feature engineering. Having consolidated a few steps of the kill chain, the MLBSA methodology consists of five phases, which are organized in a circular form implying a continuous and incremental process. The five phases of the MLBSA [12] methodology are 1) reconnaissance, 2) data collection, 3) feature engineering, 4) attacking the objective, and 5) evaluation. The following subsections will illustrate each phase in details.

Cyber–physical systems (CPSs) are complex systems that involve technologies such as control, communication, and computing. Nowadays, CPSs have a wide range of applications in smart cities, smart grids, smart manufacturing and intelligent transportation. [13],[14],[15] However, with integration of industrial control systems with modern communication technologies, CPSs would be inevitably exposed to increasing security threats, which could lead to severe degradation of the system performance and even destruction of CPSs. This paper presents a survey on recent advances on security issues of industrial cyber–physical systems (ICPSs).

3. PROBLEM STATEMENT

There are two distinct categories of cyber-attack detection methods, namely signature based and anomaly based. The machine learning techniques are used in both of them. Recently machine learning-based algorithms have been used for developing signatures that will efficiently identify both the code and behaviour of the malicious code.

The Network-based Signature Generation (NSG) [6], Length-based Signature Generation (LSEG) [7] and F-Sign [8] are the examples of algorithms designed for automated and fast extraction of signatures of polymorphic worms.

The LESG algorithm targets those worms that use buffer overflow attack to infect victims, whereas the F-Sign extracts the signature on a basis of the code of a worm (such signature can be used to detect and stop the worm from spreading). In literature there are also algorithms such as SA (Semantic Aware [9]) that are designed to generate the signatures of malicious software on a basis of the network traffic they generate. Such solutions can even properly identify malicious behavior when the traffic is noise-like [9].

The anomaly based methods for cyber-attack detection typically build a model that describes normal and abnormal behaviour of network traffic. Commonly, such methods use three types of algorithms taken from machine learning theory, namely unsupervised, semi-supervised and supervised.

For unsupervised learning often clustering approaches are used that usually adapt algorithms like k -means, fuzzy c -means, QT and SVM [10–12]. The clustered network traffic established using the mentioned approaches commonly requires the decision whenever given cluster should be indicated as malicious or not. Pure unsupervised algorithms use a majority rule telling that only the biggest clusters are considered normal. That means that network events that happen frequently have no symptoms of the attack. In practice, it is a human role to tell which cluster should be considered as an abnormal one.

The supervised machine learning techniques require at least one learning phase to establish the traffic model. The learning is typically off-line and is conducted on the specially prepared (cleaned) traffic traces.

4. PROPOSED SYSTEM

The proposed method adapts machine learning paradigm. During the learning phase the labelled data is required to establish the model parameters of the normal application behaviour.

We propose to use a graph-based approach to build a set of regular expressions that model the normal HTTP requests [15] sent by client to the web application.

In the proposed approach, the segmentation components S are the regular expressions further explained in Section 3.3. In other words, our goal is to group the similar HTTP requests [18] and represent them with a single pattern. In fact, the algorithm is not only limited to the HTTP protocol and can be easily adapted to other kinds of textual data, like different kinds of log files generated by the Application or databases.

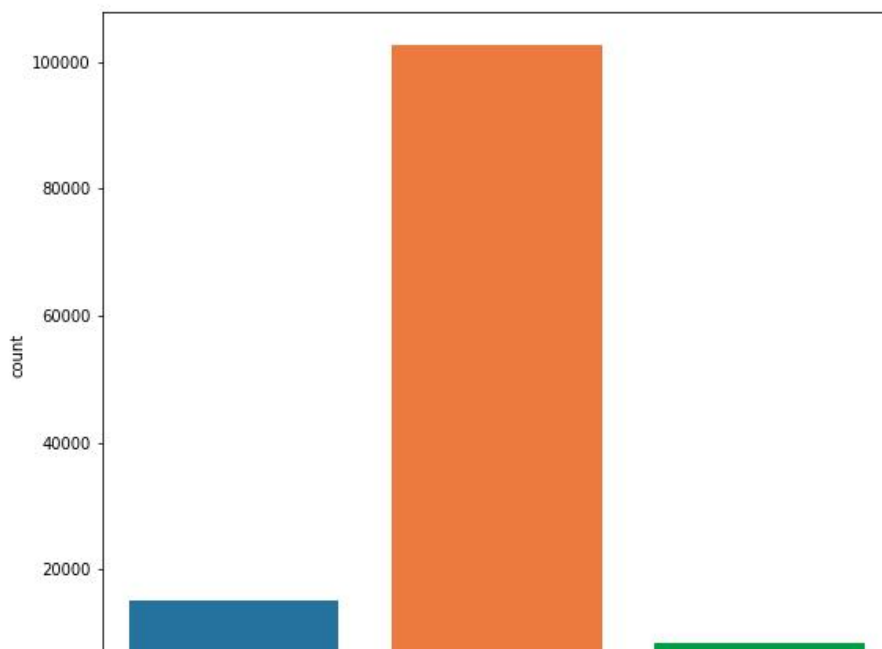
5.IMPLEMENTATION

1. DATA PREPROCESSING
2. MODEL BUILDING
3. LOGISTIC REGRESSION
4. RANDOM FOREST
5. SUPPORT VECTOR MACHINE
6. PREDICT ATTACK

5. EXPECTED OUTCOMES

Data EDA

```
: # Protocol type distribution  
plt.figure(figsize=(9,8))  
sns.countplot(x="protocol_type", data=train)  
plt.show()
```



Model Building

```
train_X=train_new[cols]  
train_y=train_new['attack_class']  
test_X=test_new[cols]  
test_y=test_new['attack_class']
```

ML Deploy

Logistic Regression

```
# Building Models
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression(random_state=0,solver='lbfgs',multi_class='multinomial')
logreg.fit( train_X, train_y)
logreg.predict(train_X)  #by default, it use cut-off as 0.5
```

```
list( zip( cols, logreg.coef_[0] ) )
```

```
logreg.intercept_
```

```
logreg.score(train_X,train_y)
```

Decision Trees

```
train_X.shape
```

```
param_grid = {'max_depth': np.arange(2, 12),
              'max_features': np.arange(10,15)}
```

```
train_y.shape
```

```
from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeClassifier, export_graphviz, export
tree = GridSearchCV(DecisionTreeClassifier(), param_grid, cv = 10,verbose=1,n_jobs=-1)
tree.fit( train_X, train_y )
```

```
tree.best_score_
```

```
tree.best_estimator_
tree.best_params_
```

```
train_pred = tree.predict(train_X)
```

```
print(metrics.classification_report(train_y, train_pred))
```

```
test_pred = tree.predict(test_X)
```


Random Forest

```

: from sklearn.ensemble import RandomForestClassifier
  pargrid_rf = {'n_estimators': [50,60,70,80,90,100],
               'max_features': [2,3,4,5,6,7]}

: from sklearn.model_selection import GridSearchCV
  gscv_rf = GridSearchCV(estimator=RandomForestClassifier(),
                        param_grid=pargrid_rf,
                        cv=10,
                        verbose=True, n_jobs=-1)

  gscv_results = gscv_rf.fit(train_X, train_y)

: gscv_results.best_params_

: gscv_rf.best_score_

: radm_clf = RandomForestClassifier(oob_score=True,n_estimators=80, max_features=5, n_jobs=-1)
  radm_clf.fit( train_X, train_y )

: radm_test_pred = pd.DataFrame( { 'actual': test_y,
                                   'predicted': radm_clf.predict( test_X ) } )

```

Support Vector Machine (SVM)

```

: from sklearn.svm import LinearSVC
  svm_clf = LinearSVC(random_state=0, tol=1e-5)
  svm_clf.fit(train_X,train_y)

: print(svm_clf.coef_)
  print(svm_clf.intercept_)
  print(svm_clf.predict(train_X))

: from sklearn.svm import SVC
  from sklearn.pipeline import make_pipeline

  model = SVC(kernel='rbf', class_weight='balanced',gamma='scale')

: model.fit(train_X,train_y)

: from sklearn.model_selection import GridSearchCV
  param_grid = {'C': [1, 10],
               'gamma': [0.0001, 0.001]}
  grid = GridSearchCV(model, param_grid)

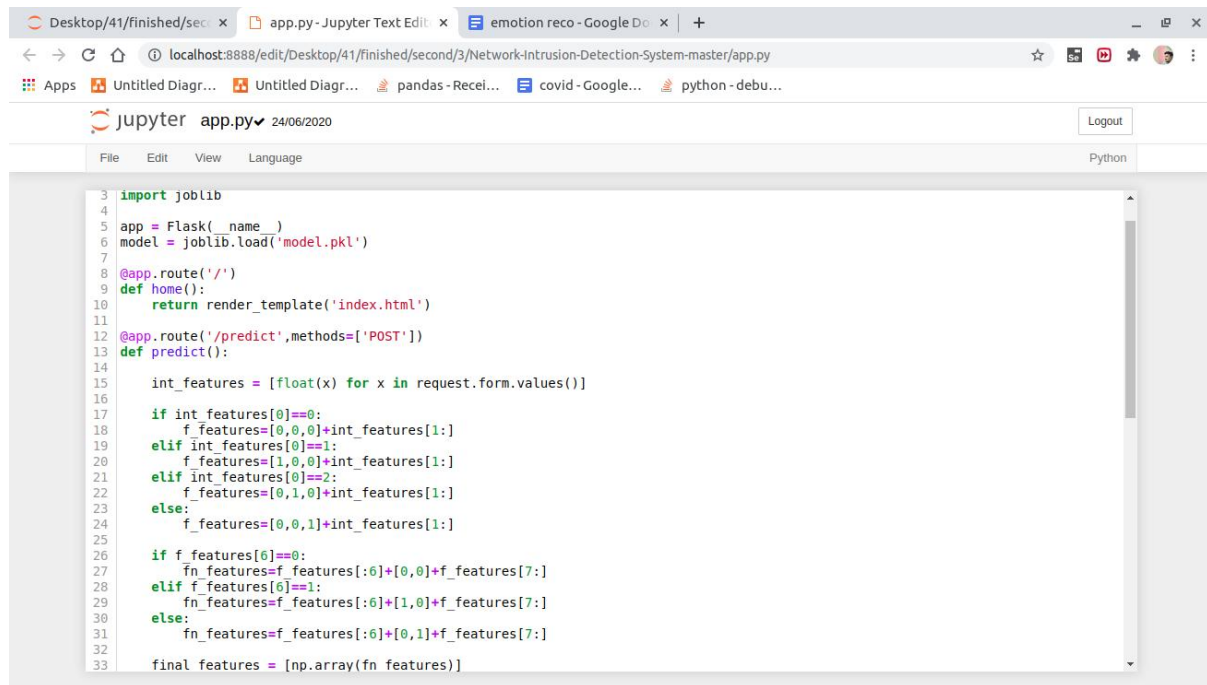
  grid.fit(train_X,train_y)

: print(grid.best_params_)

```

From the score accuracy we concluding the DT & RF give better accuracy and building pickle file for predicting the user input

Application



```

3 import joblib
4
5 app = Flask(__name__)
6 model = joblib.load('model.pkl')
7
8 @app.route('/')
9 def home():
10     return render_template('index.html')
11
12 @app.route('/predict', methods=['POST'])
13 def predict():
14
15     int_features = [float(x) for x in request.form.values()]
16
17     if int_features[0]==0:
18         f_features=[0,0,0]+int_features[1:]
19     elif int_features[0]==1:
20         f_features=[1,0,0]+int_features[1:]
21     elif int_features[0]==2:
22         f_features=[0,1,0]+int_features[1:]
23     else:
24         f_features=[0,0,1]+int_features[1:]
25
26     if f_features[6]==0:
27         fn_features=f_features[:6]+[0,0]+f_features[7:]
28     elif f_features[6]==1:
29         fn_features=f_features[:6]+[1,0]+f_features[7:]
30     else:
31         fn_features=f_features[:6]+[0,1]+f_features[7:]
32
33     final_features = [np.array(fn_features)]

```

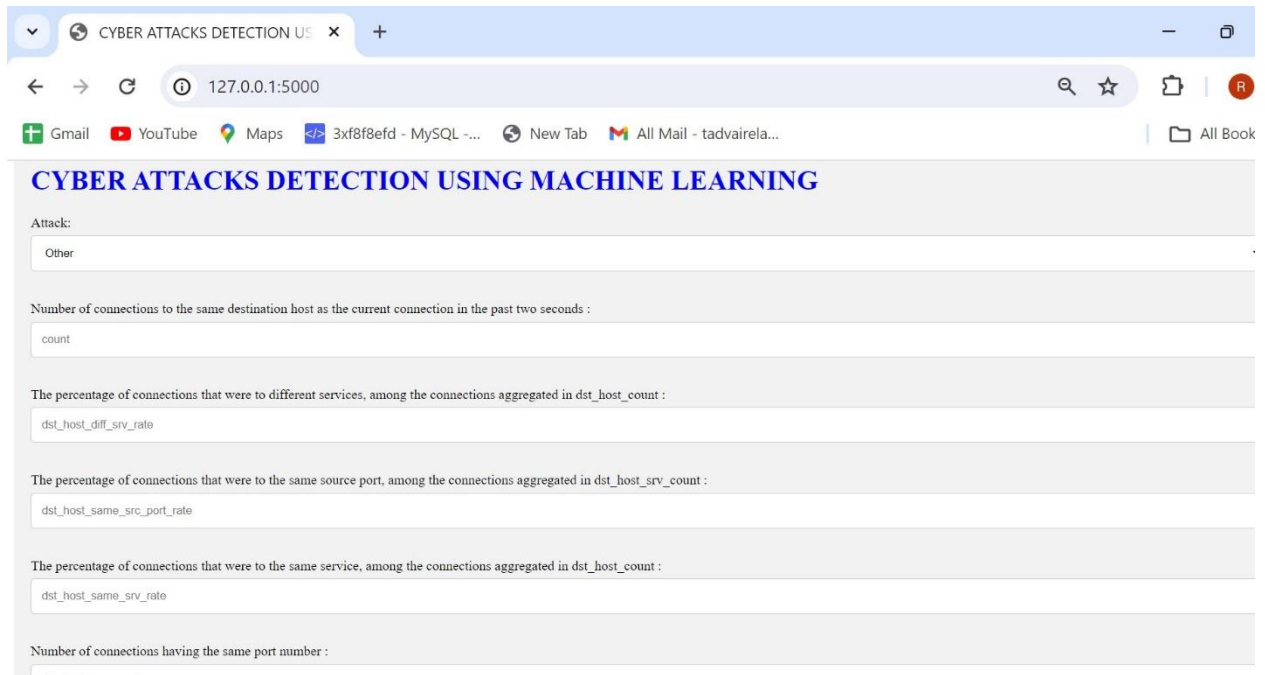
Localhost - in cmd python app.py

```

user@ramesh:~/Desktop/41/finished/second/3/Network-Intrusion-Detection-System-master$ python3 app.py
/home/user/.local/lib/python3.6/site-packages/sklearn/base.py:334: UserWarning:
Trying to unpickle estimator LogisticRegression from version 0.22.1 when using v
ersion 0.23.2. This might lead to breaking code or invalid results. Use at your
own risk.
  UserWarning)
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

```

Enter the input



The screenshot shows a web browser window with the title 'CYBER ATTACKS DETECTION USING MACHINE LEARNING'. The address bar displays '127.0.0.1:5000'. The browser's taskbar at the bottom includes icons for Gmail, YouTube, Maps, a terminal window titled '3xf8f8efd - MySQL -...', a 'New Tab', and an email client titled 'All Mail - tadvairera...'. The webpage content is as follows:

CYBER ATTACKS DETECTION USING MACHINE LEARNING

Attack:

Other

Number of connections to the same destination host as the current connection in the past two seconds :

count

The percentage of connections that were to different services, among the connections aggregated in dst_host_count :

dst_host_diff_srv_rate

The percentage of connections that were to the same source port, among the connections aggregated in dst_host_srv_count :

dst_host_same_src_port_rate

The percentage of connections that were to the same service, among the connections aggregated in dst_host_count :

dst_host_same_srv_rate

Number of connections having the same port number :

Status of the connection –Normal or Error :

Other

Last Flag :

3

1 if successfully logged in; 0 otherwise :

5

The percentage of connections that were to the same service, among the connections aggregated in count :

5

The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in count :

2

Destination network service used http or not :

No

Predict

5 : 2024

7. CONCLUSION

USION

In this article, the method for application layer attack detection based on machine learning was proposed. The model consists of patterns (in form of PCRE regular expressions) that are obtained using graph-based segmentation technique and dynamic programming. The regular expressions are used for modelling the genuine behaviour of the applications and detecting cyber attacks.

We also presented the results that prove the efficiency of the proposed algorithm that can be effectively used for application layer attack detection.

The experiments on CSIC'10 show that the proposed approach can achieve 94.46% of detection ratio while having <4.5% of false positives.

8.FUTURE SCOPE

The future scope of Cyber-attack Detection using techniques like Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN) is quite promising. Here are some potential areas of development and application:

- Enhanced Detection Accuracy:** Continuous improvement in machine learning algorithms and techniques can lead to better detection accuracy. This involves refining feature selection, optimizing model parameters, and incorporating more advanced algorithms such as deep learning and reinforcement learning.

2. **Adversarial Attack Detection:** As cyber attackers become more sophisticated, they may attempt to evade detection by crafting attacks specifically designed to bypass machine learning-based detection systems. Future research will focus on developing robust models capable of detecting such adversarial attacks.
3. **Real-time Threat Intelligence Integration:** Integrating machine learning models with real-time threat intelligence feeds can enhance detection capabilities by providing up-to-date information about known threats and attack vectors.
4. **Privacy-preserving Techniques:** As data privacy concerns grow, there will be increased emphasis on developing techniques that allow for effective cyber-attack detection while preserving the privacy of sensitive information.
5. **Interdisciplinary Approaches:** Cybersecurity is a multidisciplinary field, and future advancements may come from combining insights and techniques from fields such as psychology, sociology, and economics to better understand attacker motivations and behaviours.

The future of cyber-attack detection using machine learning techniques holds promise for improved accuracy, scalability, and adaptability to evolving threats. Continued research and development in this area will be crucial for staying ahead of cyber adversaries.

9. REFERENCES

- [1] CERT Polska Annual Report 2012. http://www.cert.pl/PDF/Report_CP_2012.pdf
- [2] SOPHOS homepage <http://www.sophos.com>
- [3] Cisco Annual Report 2013. http://www.cisco.com/web/about/ac49/ac20/ac19/ar2013/docs/2013_Annual_Report.pdf
- [4] BYOD: Bring Your Own Device. <http://www.vs.inf.ethz.ch/publ/papers/rohs-byod-2004.pdf>
- [5] OWASP Top 10 2013. https://www.owasp.org/index.php/Top_10_2013-Top_10
- [6] NSG. <http://www.ijcst.com/vol31/4/sridevi.pdf>

- [7] LESG. <http://www.cs.northwestern.edu/~ychen/Papers/LESG-ICNP07.pdf>
- [8] A. Shabtai, E. Menahem and Y. Elovici. F-Sign: automatic, function-based signature generation for malware, systems, man, and cybernetics, Part C: applications and reviews. Transactions on IEEE, 41, 494–508, 2011.
- [9] D. Kong, J. Gong, S. Zhu, P. Liu and H. Xi. SAS: semantics aware signature generation for polymorphic worm detection. International Journal of Information Security, 50, 1–19, 2011.
- [10] M. Sharma and D. Toshniwal. Pre-clustering algorithm for anomaly detection and clustering that uses variable size buckets. Recent Advances in Information Technology, 515–519, 2012.
- [11] M. H. A. C. Adaniya, M. F. Lima, J. J. P. C. Rodrigues, T. Abrao and M. L. Proenca. Anomaly detection using DSNS and FireflyHarmonic Clustering Algorithm. Communications (ICC), 1183–1187, 2012.
- [12] J. Mazel, P. Casas, Y. Labit and P. Owezarski. Sub-space clustering, Inter-Clustering Results Association and anomaly correlation for unsupervised network anomaly detection. Network and Service Management (CNSM), 1–8, 24–28 October 2011.
- [13] Kaspersky Lab. Security report. <http://www.securelist.com/en/analysis/204792244/Thegeography-of-cybercrime-Western-Europe-and-North-America>
- [14] ESET threat report 12-2012. <http://go.eset.com/us/resources/threat-trends/Global-ThreatTrends-November-2012.pdf>
- [15] F. Felzenszwalb and P. Huttenlocher. Efficient graph-based image segmentation. International Journal of Computer Vision, 59, 167–181, September 2004.
- [16] B. Needleman Saul and D. Wunsch Christian A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology, 48, 443–453, 1970.
- [17] CSIC 2010 HTTP Dataset in CSV format. http://users.aber.ac.uk/pds7/csic_dataset/csic_2010http.html

- [18] Z. Zhang, J. Li , C. Manikopoulos, J. Jorgenson and J. Ucles. HIDE: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification. In Proceeding of IEEE Workshop on Information Assurance and Security, 2001.
- [19] Adetunmbi Adebayo O., Falaki Samuel O., Adewale Olumide S. and K. Boniface. Network intrusion detection based on rough set and k-nearest neighbour. International Journal of Computing and ICT Research, 2, 60–66, 2008. Downloaded from <https://academic.oup.com/jigpal/article-abstract/23/1/45/2893005> by Open University of Hong Kong user on 23 January 2019[11:10 30/12/2014 jzu038.tex] Paper Size: a4 paper Job: JIGPAL Page: 56 45–56 56 Machine learning techniques to detect cyber attacks
- [20] J. Ma and G. ZhongXu. Network anomaly detection using dissimilarity-based one-class SVM classifier. ICPPW '09. International Conference on Parallel Processing Workshops, 2009, 409–414, 22–25 September 2009.