Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 04-2023
DOI 10.17605/OSF.IO/QYVAH

DETECTION OF MALWARE USING MACHINE LEARNING TECHNIQUES

HARITHA RAJEEV
Research scholar, Department of Information Technology, Lincoln University College, Malaysia.
Email: hrajeev@lincoln.edu.my

Dr. MIDHUN CHAKKARAVARTY
Assistant Professor, Department of Information Technology, Lincoln University College, Malaysia.
Email: midhun@lincoln.edu.my

Abstract

An application created specifically to infiltrate and harm PCs without the owner's consent is known as a malicious PC programme. Malware can take on many different shapes, such as infections, rootkits, key loggers, worms, Trojan horses, spyware, ransomware, secondary gateways, bots, logic bombs, and so forth. The amount, variety, and speed of non-PC applications' transmission are all steadily growing. The advancement of exchanging frameworks is a result of neural network concept, which has its roots in computerized reasoning. In the field of financial industry, the neural network application performs different roles like time series forecasting, algorithmic trading, securities classification, credit risk modelling, and the production of ownership indicators and exit prices, among others. The vector component and the stage delivery computation both have an impact. They also coordinate both include based on substance and include based on behavior. They put out the SVM-AR integrated learning approach, which combines hierarchical principles and a vector backing machine.

***Keywords*:** Random forests, Logistic Regression, Neural Network

1. INTRODUCTION

The main objective of the designed malicious computer program is to damage the various operations of the computer system. This malware recognition framework is used to decide whether a program contains malicious content or not. The motivation to create a non-computer program has changed dramatically in recent years. Most of the authors of the malware program were financially motivated. In detecting a malware-based computer-based program, the virus protection program detects an invalid signature without a byte sequence in a specific file to declare the content as pernicious In polymorphic infections and obscure infections, the mark based identification framework fizzles in light of the fact that polymorphic infections are infected and change the descriptor circle in every disease without changing the genuine code and in obscure infections no mark exists in the antivirus data set. Anomaly-based system detects any computer misuse from the normal computer function. An unusually based acquisition system monitors a PC program or a few programming movement and groups it as typical or stomach muscle typical. Social based discovery framework, recognizes an activity performed by a malware. A machine-based identification system requires a training data set for malware attributes and as a result the AI calculation distinguishes a malware program. There are different machine calculations like Choice Tree, Backing Vector Machine, Arbitrary Forest, Supporting, and so on Obscurity methods, for

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 04-2023
DOI 10.17605/OSF.IO/QYVAH

example, dead code coding, Vault Decrease, Code Move, Order Changes upset the most common way of finding a vindictive PC program.

2. RELATED WORK

Explain a simple behavioral malware detection method that makes use of prefetch files in Windows. We use two different Windows platforms and a variety of programmes to show how our malware detection is generalized. We examine the performance degradation of our malware detection due to idea drift and its flexibility. [1]. the most popular operating system for upcoming smart devices is now Android. As a result, Android malware has increased dramatically. To find Android malware, many dynamic analysis methods have been presented. Malware detection framework that can support various detectors from outside sources (such as researchers and antivirus providers) and permits effective and in-control real-time monitoring. [2]. the exponential proliferation of malware in this digital age is a major security risk for computer users, businesses, and governmental agencies. Static and dynamic analysis, which are used in conventional malware detection techniques, are useless for identifying unidentified malware. By utilizing polymorphic and evasion tactics on already-existing malware to avoid detection, malware developers create new malware. The vision-based approach can be used to examine the patterns of recently introduced malware, which are variations of already existing malware. Malware patterns are represented as images, and their characteristics are listed. In order to categories malware, an alternate generation of class vectors and feature vectors employing ensemble forests in many sequential layers is carried out. In contrast to deep learning models, this work presents a hybrid stacked multilayered ensembling technique [3]. Although malware and its variations may differ significantly from content signatures, we have found that they do share several higher-level behavioural traits that are more accurate in exposing the infection's primary goal. In some case malware detection algorithm based on the extraction of malicious behaviour features, (MBF) extraction strategy, and the method of extracting malware behaviour are all investigated. After being created and put into operation, malware detection system based on MBF has shown through testing results that it is capable of detecting freshly found unidentified malwares. [4]. The convolutional neural network (CNN) and other machine learning methods are used by the authors of this paper to provide an automated malware detection method. These days, malware detection technologies are vulnerable to malware with erratic port numbers and protocols since they depend on the programmes' chosen packet fields, such as the port number and protocols. Since the suggested method leverages 35 different features collected from packet flow rather than ports and protocols, it offers more reliable and accurate malware detection [5]. evaluating the Stratosphere IPS project's data. The number of people using mobile devices is growing rapidly. Because of this, machine learning techniques should be used to create automated malware screening solutions [6]. Because of the rapidly evolving technologies, malware experts have always had a difficult time detecting malware. We use machine and deep learning (ML/DL) to enhance the detection process in order to combat the advancements in modern viruses, which even have the ability to bring down large organizations in a matter of seconds.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 04-2023
DOI 10.17605/OSF.IO/QYVAH

Without manual intervention, which frequently results in missing sophisticated or disguised malware, this aids in the efficient and precise detection of malware. In order to further analyses, the virus, this article aims to train and evaluate a variety of deep learning neural networks, including convolutional and recurrent networks. Also, we do a comparison analysis using other performance criteria. With CNN, we were able to attain validation accuracy of over 98%. [7]. The Convolutional Neural Network (CNN)) has been successful in a variety of disciplines, including the identification of Android malware. Yet, it has been demonstrated through empirical analysis of earlier experiments that no single machine learning classifier can deliver the highest accuracy in any circumstance. In order to increase the effectiveness of malware classification [8]. Efficient malware identification is a current necessity across all sectors. It is essential to invest in creating systems to detect and quarantine malware as sophisticated polymorphic malware spreads. With the use of static analysis and machine learning, this work attempts to address the issue of malware detection [9]. Malicious software, also referred to as malware, is intentionally created to disrupt networks and inflict harm in a number of different ways. The main objective of malware is to obstruct regular operation or to gain unauthorized access.A malicious application impersonates a common program[10]. A virus is now only detected by commercial anti-virus software after it has already manifested and harmed data. Our system is built on fuzzily inferred behavior from malicious code. More abstract descriptions of malware are produced thanks to the decompile approach, which describes the structural and behavioural properties of binary code. The recommended solution can automatically acquire the fuzzy subsets and their membership function thanks to the GD-FNN learning algorithm [11]. Hackers have been using covert network attacks in recent years to break into systems and steal high-value information. These attacks are motivated by commercial and political motives and leverage social engineering techniques and system vulnerabilities. Traditional detection technologies cannot properly detect, track, and evaluate it because of the usage of these cutting-edge technology and long-term delay strategies. Long-term delay, recurrent connections, and other features are characteristics of concealed network attacks [12]. Cyberattacks pose a serious threat to the distribution of electricity as the energy internet develops. Particularly, rogue domains can seriously undermine network security. The problem of identifying DGA-generated domains has been attempted by numerous academics using various methods. To solve this issue, a Deep Bidirectional LSTM model-based algorithm is built in this paper [13]. The importance of cyber security has increased recently. It becomes difficult to determine how to recognize suspected malware. We use deep learning techniques and run flow detection on actual data to address this problem. Unfortunately, the uneven data distribution that real data frequently experiences will result in a gradient dilution problemThis problem highlights the neural network's bias during training towards the majority class and its inability to pick up knowledge from the minority classes. To identify the data layer by layer, the authors of this research propose TSDNN model [14]. Malware is increasingly being distributed through email attachments. Although though machine learning (ML) has been successfully used to detect malware in portable executables (PE) [15].

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 04-2023
DOI 10.17605/OSF.IO/QYVAH

Feature Extraction

Feature Extraction implies choosing subset of element from entirety. Highlight ve-peak assumes significant part in building AI model. Head Component Analysis (PCA) and Feature Rank calculation has been utilized to ex-lot most significant highlights. Exploring the malware dataset from githud. It consist of 41323 legitimate files and 96724 malware files. Using describe function to describe the entire dataset.in the case of target variable see that the 0 which represent the malware file are around 6000 something 1 which represent the original files around 4500 something

3. EXPERIMENT SETUP AND WORKING

The accompanying devices Algorithms are expected for exploratory arrangement are port-capable executable records Random Forest, Logistic Regression, Neural Network. Using depict capacity to portray the whole dataset.

For smart working remove the missing values, In this case there is no missing values.so it does not gives any signals, Then split the data in to test and train. sklearn model selection is important for splitting data

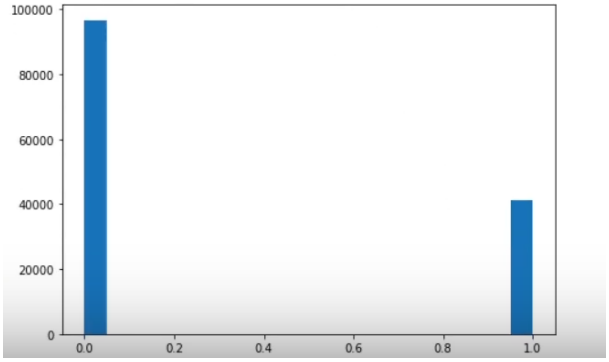

Fig 1: Shape of Legit and Malware dataset

The shape of legit dataset is: 41321samples 56 features

The shape of Mal dataset is: 96724 samples 56 features

3.1. Random forests

Random forests are a coordinated learning technique for isolation, recovery and different exercises that work by building various deciduous trees during preparing. With isolation exercises, random forest clearing is a class chosen by many trees. With retrospective operations, the average rate or forecast for each tree is reversed. Random forests often make better decisions, but their accuracy is lower than that of improved trees. However, data features can affect their performance

The purpose of these two random springs is to reduce the variability of the forest scale. Indeed, trees of each decision tend to show high diversity and are often overcrowded. Randomly injected into the woods produces decision trees with guessing errors that are somehow separated. By taking a look at those predictions, some errors can be

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 04-2023
DOI 10.17605/OSF.IO/QYVAH

canceled. Random forests gain reduced diversity by combining a variety of trees. In fact the variance reduction is often noticeable and therefore provides a much better model.

the use of scikit-learn involves class dividers by estimating their probability, rather than allowing each category to vote for one category.

For this import Random forest classifier, by using sklearn and for evaluation using make classification.

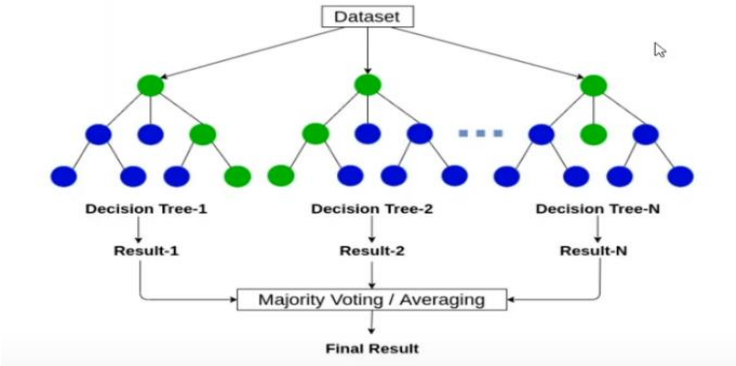

Fig 2: Random Forest

After applying random forest it will give the result as

Accuracy on the train dataset=.9828

Accuracy on test dataset=0.9838

	Predicted Yes	Predicted No
Actual Yes	TP= 19080	FP=170
Actual NO	FN=277	TN=8083

Fig 3: Confusion matrix of Random Forest

3.2. Logistic Regression

Logistic Regression is one of the most famous techniques for AI, which goes under the management of a regulated learning procedure. It is utilized to foresee stage subordinate changes utilizing a given arrangement of free factors.Relapse predicts the surge of stage subordinate changeability. In this manner the outcome ought to be stage or separate worth. Either Yes or No, 0 or 1, Valid or Bogus, etc yet as opposed to giving a prompt worth, for instance, 0 and 1, it gives potential characteristics some place in the scope of 0 and 1.

Logistic Regression is basically the same as Linear Regression paying little mined to the way things are utilized. Linear Regression is utilized for troubleshooting issues, and Logistic relapse is utilized for troubleshooting issues.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 04-2023
DOI 10.17605/OSF.IO/QYVAH

In Calculated relapse, rather than embedding a relapse line, we are equivalent to the "S" formed altering capability, which predicts two higher qualities (0 or 1). a one-versus rest (OvR) plot when the 'multi_class' decision is set to 'ovr', and uses cross-entropy incident if the 'multi_class' decision is set to 'multinomial'. '. (At present the 'multinomial' decision is only maintained for 'Albbgs', 'hang', 'experience' and 'newton-cg' courses of action.). This portion applies the standard backslide using the library 'liblinear', 'newton-cg', 'hang', 'experience' and 'lbfgs' game plans

It can manage both thick and little wellsprings of data. Use comparable C-mentioned people or CSR grid containing 64-bit floats for ideal execution; some other data game plan will be changed over (and copied).

Deals with serious consequences regarding 'newton-cg', 'hang', and 'lbfgs' simply help L2's information on basic turn of events, or none using any and all means. The 'liblinear' plan maintains both the L1 and

L2 plans, with only two L2 balance improvements. Versatile Net suspension is just upheld by the '' adventure 'arrangement

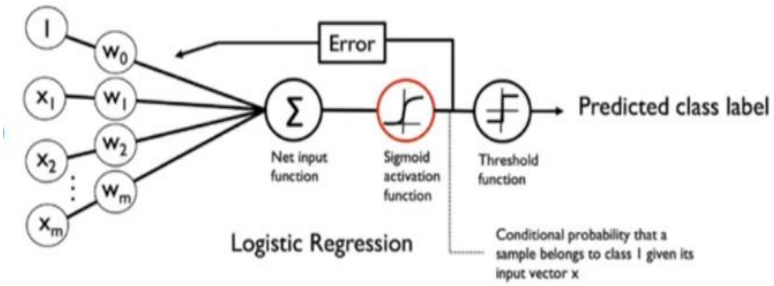

Fig 4: Logistic Regression

After applying random forest it will give the result as

Accuracy on the train dataset=. o.70152

Accuracy on test dataset=0.6972

	Predicted Yes	Predicted No
Actual Yes	TP= 19250	FP=0
Actual NO	FN=8360	TN=0

Fig 5: Confusion matrix of Logistic Regression

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 04-2023
DOI 10.17605/OSF.IO/QYVAH

3.3 *Neural Network*

A Neural network is a series of calculations that imitates the functions of the human frontal brain in an effort to identify fundamental links among a massive amount of data. In this way, neural networks in the brain indicate neuron designs that can either be produced by external influences or occur naturally. Without needing to alter the result conditions, neural organisations are the most effective at adapting to new circumstances and producing the greatest outcomes. The advancement of exchanging frameworks is a result of the concept of neural networks, which has its roots in computerised reasoning. The vector component and the stage delivery computation both have an impact.. They also coordinate both include based on substance and include based on behavior. They put out the SVM-AR integrated learning approach, which combines hierarchical principles and a vector backing machine.

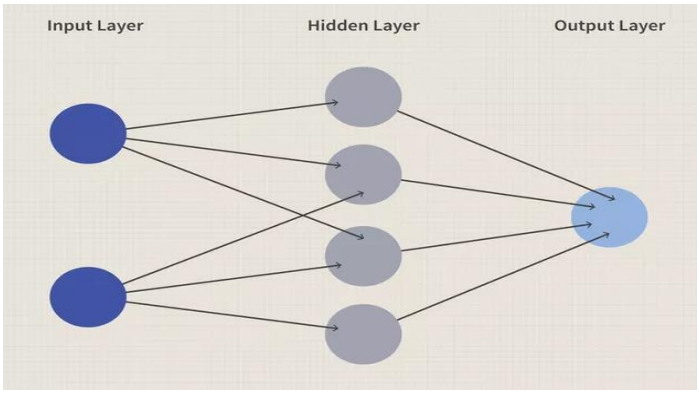

Fig 6: Neural Network

Multi-layer perceptron (MLP) estimate is carried out by Class MLPClassifier utilising backpropagation training.

MLP trains on two displays: display y of size (n samples,), which carries the goal characteristics, and bunch X of size (n samples, n features), which holds the arrangement tests commonly referred to as floating point integrate vec-pinnacles.

Layer (type)	Output Shape	Param #
dense_28 (Dense)	(None, 16)	880
dense_29 (Dense)	(None, 8)	136
dense_30 (Dense)	(None, 4)	36
dense_31 (Dense)	(None, 1)	5

Total params: 1,057
Trainable params: 1,057
Non-trainable params: 0

Fig 7: Layers in Neural Network

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 04-2023
DOI 10.17605/OSF.IO/QYVAH

>>> from sklearn.neural_network import MLPClassifier

Import TensorFlow. Then define the model and compile it,In neural network it is not a complete network

Evaluate the model and finding accuracy on trained and test data

Accuracy on the train dataset=.0.9538

Accuracy on test dataset=0.9538

	Predicted Yes	Predicted No
Actual Yes	TP= 19030	FP=220
Actual NO	FN=1055	TN=7305

Fig 8: Confusion matrix of Neural Network

Models	Train Data Accuracy	Test Data Accuracy	F1-Score	Position
Random Forest	0.98%	0.98%	0.97%	First
Logistic Regression	0.70%	0.69%	0.00%	Third
Neural Network	0.95%	0.95%	0.91%	Second

Fig 9: Model performance comparison

CONCLUSION

The revelation of malware is filling in the exploration region because of worries about the everyday expansion in malware. Signature-based antivirus framework isn't valuable for recognizing malware and they are encountering troubles due to polymorphic infections and multi day assaults. Signature-based strategies ought to along these lines be joined by another technique that can distinguish unknown malware. A pernicious PC program is a PC program or piece of programming that is expected to attack and destroy computers without the owner's assent. There are various kinds of malware program, for instance, infections, rootkits, keyloggers, worms, trojan, spyware, ransomware, secondary passages, bots, rationale bombs, etc. The volume, change and speed of transport of non-PC programs are growing each year. Random forest works

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 04-2023
DOI 10.17605/OSF.IO/QYVAH

better in malware of dataset that is the explanation this not generally around acted in this association

References

1. B. Alsulami, A. Srinivasan, H. Dong and S. Mancoridis, "Lightweight behavioral malware detection for windows platforms," *2017 12th International Conference on Malicious and Unwanted Software (MALWARE)*, Fajardo, PR, USA, 2017, pp. 75-81, doi: 10.1109/MALWARE.2017.8323959.
2. S. Iqbal and M. Zulkernine, "SpyDroid: A Framework for Employing Multiple Real-Time Malware Detectors on Android," *2018 13th International Conference on Malicious and Unwanted Software (MALWARE)*, Nantucket, MA, USA, 2018, pp. 1-8, doi: 10.1109/MALWARE.2018.8659365.
3. S. A. Roseline, A. D. Sasisri, S. Geetha and C. Balasubramanian, "Towards Efficient Malware Detection and Classification using Multilayered Random Forest Ensemble Technique," *2019 International Carnahan Conference on Security Technology (ICCST)*, Chennai, India, 2019, pp. 1-6, doi: 10.1109/CCST.2019.8888406.
4. W. Liu, P. Ren, K. Liu and H. -x. Duan, "Behavior-Based Malware Analysis and Detection," *2011 First International Workshop on Complexity and Data Mining*, Nanjing, China, 2011, pp. 39-42, doi: 10.1109/IWCDM.2011. 17..
5. M. Yeo *et al*., "Flow-based malware detection using convolutional neural network," *2018 International Conference on Information Networking (ICOIN)*, Chiang Mai, Thailand, 2018, pp. 910-913, doi: 10.1109/ICOIN.2018.8343255.
6. B. TAHTACI and B. CANBAY, "Android Malware Detection Using Machine Learning," *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Istanbul, Turkey, 2020, pp. 1-6, doi: 10.1109/ASYU50717.2020.9259834.
7. Moskovitch R, Feher C, Elovici Y. Unknown malcode detection—a chronological evaluation. H. Malani, A. Bhat, S. Palriwala, J. Aditya and A. Chaturvedi, "A Unique Approach to Malware Detection Using Deep Convolutional Neural Networks," *2022 4th International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)*, KualaLumpur, Malaysia, 2022, pp. 1-6, doi: 10.1109/ICECIE55199.2022.10000344.
8. Y. Jin, T. Liu, A. He, Y. Qu and J. Chi, "Android Malware Detector Exploiting Convolutional Neural Network and Adaptive Classifier Selection," *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, Tokyo, Japan, 2018, pp. 833-834, doi: 10.1109/COMPSAC.2018.00143.
9. K. Thosar, P. Tiwari, R. Jyothula and D. Ambawade, "Effective Malware Detection using Gradient Boosting and Convolutional Neural Network," *2021 IEEE Bombay Section Signature Conference (IBSSC)*, Gwalior, India, 2021, pp. 1-4, doi: 10.1109/IBSSC53889.2021.9673266.
10. K. Gupta, N. Jiwani, M. H. U. Sharif, R. Datta and N. Afreen, "A Neural Network Approach for Malware Classification," *2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, Greater Noida, India, 2022, pp. 681-684, doi: 10.1109/ICCCIS56430.2022.10037653.
11. Y. Zhang, J. Pang, F. Yue and J. Cui, "Fuzzy Neural Network for Malware Detect," *2010 International Conference on Intelligent System Design and Engineering Application*, Changsha, China, 2010, pp. 780-783, doi: 10.1109/ISDEA.2010.314.
12. J. Nie, P. Ma, B. Wang and Y. Su, "A Covert Network Attack Detection Method Based on LSTM," *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Chongqing, China, 2020, pp. 1690-1693, doi: 10.1109/ITOEC49072.2020.9141848.

Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 42 Issue: 04-2023
DOI 10.17605/OSF.IO/QYVAH

13. Y. Liang and X. Yan, "Using Deep Learning to Detect Malicious URLs," *2019 IEEE International Conference on Energy Internet (ICEI)*, Nanjing, China, 2019, pp. 487-492, doi: 10.1109/ICEI.2019.00092.
14. Y. -C. Chen, Y. -J. Li, A. Tseng and T. Lin, "Deep learning for malicious flow detection," *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Montreal, QC, Canada, 2017, pp. 1-7, doi: 10.1109/PIMRC.2017.8292316.
15. E. M. Rudd, R. Harang and J. Saxe, "MEADE: Towards a Malicious Email Attachment Detection Engine," *2018 IEEE International Symposium on Technologies for Homeland Security (HST)*, Woburn, MA, USA, 2018, pp. 1-7, doi: 10.1109/THS.2018.8574202.