Bespoke Sequence of Transformations for an Enhanced Entropic Wavelet Energy Spectrum Discernment for Higher Efficacy Detection of Metamorphic Malware

A Nonnegative Matrix Factorization, Multiresolution Matrix Factorization, and Continuous Wavelet Transform Amalgam

Steve Chan
Decision Engineering Analysis Laboratory, VTIRL, VT
Orlando, USA
e-mail: schan@dengineering.org

***Abstract*—A Robust Convex Relaxation (RCR) Long Short-Term Memory (LSTM) Deep Learning Neural Network (DLNN) can provide enhanced Entropic Wavelet Energy Spectrum (EWES) discernment regarding the potential use of packers, crypters, and protectors (it has been found that compressed or encrypted files have greater entropy values), which can be indicative of Metamorphic Malware (MM). The RCR-LSTM DLNN facilitates a more robust Recurrent Neural Network (RNN) to Feedforward Neural Network (FNN) progression via a bespoke Nonnegative Matrix Factorization (NMF) to Multiresolution Matrix Factorization (MMF) to Continuous Wavelet Transform (CWT) Sequence of Transformations (SOT). Preliminary experimentation pertaining to the RCR-LSTM DLNN framework indicates potential higher efficacy for an enhanced EWES discernment than traditional Machine Learning (ML) and DLNN methods. The potential impact includes the greater use of Industrial Internet of Things (IIOT) sensors, which have been beset by MM, for Industrial Control Systems (ICS), among others.**

Keywords-Industrial Systems; Industrial Control Systems; Distributed Control Systems; Operational Technology; Condition Monitoring Paradigm; Industrial Internet of Things; Metamorphic malware.

I. INTRODUCTION

The need for a greater volume and variety of sensors within the Operational Technology (OT) ecosystem — with higher resolution and enhanced edge analytics — has been steadily increasing over the past decade. As just one example, the involved Operation and Maintenance (O&M) Condition Monitoring Paradigm (CMP) is often established by policy in a top-down fashion and may be uniform throughout a region (without considering the greatly varied ambient factors affecting the locales); by way of example, Region A may be subject to seismic activity, Region B by high wind and salinity, Region C by heavy rainfall, high humidity, and lightning, and Region D by drought and high temperatures. Intuitively, the CMPs should be tailored to fit the regions accordingly, but quite frequently, this is not the case. As the equipment in these varied areas have experienced faster than anticipated degradation and failure rates, the introduction of specialty sensors to detect for aberrant conditions has become paramount. Yet, the use of such Industrial Internet of Things (IIOT) sensors are also beset by an array of potential cyber-related vulnerabilities, which has hindered their deployment and utilization. In particular, there has been a surge in polymorphic and metamorphic malware in this arena. If timely patching — which is often difficult in numerous OT environs that have high uptime requirements — is problematic, then alternative mitigation pathways are quite limited. Along this particular vein, the study space is still, comparatively, fairly nascent.

This paper posits that an amalgam of Nonnegative Matrix Factorization (NMF), Multiresolution Matrix Factorization (MMF), and Continuous Wavelet Transform (CWT) can be of some value-added proposition in MM discernment. This amalgam, particularly with regards to the Numerical Implementation (NI) of CWT, was operationalized via a particular class of Convolutional Neural Networks (CNNs) — a RCR-based Convolutional LSTM DLNN, which leverages deeper cascade learning (thereby nicely emulating CWTs). In addition to its value-added proposition of convex relaxation adversarial training, the RCR-LSTM DLNN framework also enhances the bounds tightening for the successive convolutional layers (which contain the cascading of ever-smaller "CWT-like" convolutional filters) for an Enhanced Discernment Accuracy or EDA capability, via support of the facilitation for an enhanced MM EWES Discernment (M2ED).

This paper is structured as follows. Section I provides a backdrop and introduces the problem space. Section II presents relevant background information and discusses the operating environment, as well as the state of the challenge. Section III provides some theoretical foundations and the posited/utilized approach. Section IV delineates a strategy for a Sequence of Transformations (SOT) and delineates some preliminary experimental forays regarding the referenced RCR-LSTM DLNN framework. Section V concludes with some preliminary reflections, puts forth envisioned future work, and the acknowledgements close the paper.

II. BACKGROUND INFORMATION

Over the past several years, there has been a rapid convergence at the nexus of Information Technologies (IT) and OT, particularly in the realm of Industrial Systems (IS). As the requisite uptime and High Availability (HA) of various IS, such as Industrial Control Systems (ICS), Distributed Control Systems (DCS), etc. have increased, the need for an enhanced O&M CMP has also increased. This has involved the desire for a greater use of IIOT sensors, which can have higher resolution, greater reliability, and the potential for providing advance warning with regards to the potential failure of the involved devices (i.e., single item) and equipment (i.e., multiple items) within the CMP.

Legacy IS architecture has been, traditionally, bus topology-centric; this presumes that the involved devices/equipment are connected to the bus and have a common protocol. However, to leverage the wide array of specialized IIOT devices/sensors, which might not share the same protocol, REpresentational State Transfer (REST) Application Programming Interfaces (APIs) are often relied upon. IT/OT engineers have utilized REST APIs so as to obviate the need for protocol conversion, middleware, and/or gateways. These APIs are now heavily relied upon to detect issues, within IT/OT-related paradigms, such as that of unusually high temperatures, vibrations, etc. Yet, many other parameters are in need of monitoring as well. Unfortunately, many of the utilized APIs fall into the category of, among others, Open Worldwide Application Security Project (OWASP) API9:2023 and API10:2023; OWASP 9 (Improper Inventory Management) cites the use of deprecated API versions and exposed debug endpoints, and OWASP 10 (Unsafe Consumption of APIs) cites the use of potentially compromised third-party APIs.

According to a Dragos report, while 65% of advisories contain a patch to fix the cited vulnerability, it was challenging to implement the patch due to the downtime risk for the involved OT system [1]; in addition, there was no viable alternative mitigation, if patching was not an option. This paradigm is aggravated by the fact that, according a SysAdmin, Audit, Network, and Security (SANS) Institute survey, “Threat-Informed Operational Technology Defense: Securing Data vs. Enabling Physics,” “47% of ICS organizations do not have internal dedicated 24/7 ICS security response resources to manage OT/ICS incidents” [2]. Furthermore, cyberattacks are occurring with high prevalence; the World Economic Forum’s (WEF) Global Risk Report notes that these attacks on critical infrastructure operations (e.g., OT) are among the top five “currently manifesting risks” [3]. McKinsey & Company notes that OT cyberattacks have higher and more profound negative impacts, such as shutdowns, outages, and explosions [4].

Nevertheless, despite the fact that deprecated API versions and compromised third-party APIs are at play, advances in the area of mitigation have remained fairly nascent, if patching is not an option. Meanwhile, there has been an increase in the use of packers (i.e., self-extracting archives that unpack in memory upon execution of the packed file), crypters (i.e., a paradigm, wherein the use of obfuscation and/or encryption is at play), protectors (i.e., a paradigm, wherein a hybridization of both packing and encrypting is at play), etc. to obfuscate malicious intent from detectors. For example, packers greatly increase the complexity for the detectors to successfully perform statistical analysis (a prevalent approach by defenders). To aggravate matters, attackers are also anticipating the use of detection of the involved cryptor stub (i.e., a code segment or binary that accepts the malicious encrypted payload, decrypts, and executes it) signature and are now dividing the cryptor stub into multiple stages so as to obviate detection efforts. Along this vein, many attackers are now utilizing legitimate installers and supplanting the appended data with the crypter. They are also instantiating hollowed processes within trusted areas. Furthermore, they are often generating a unique binary for each compilation. Yet others utilize polymorphic code (i.e., code that utilizes a polymorphic engine to mutate its shape and signature while ensuring that the involved algorithm is preserved); indeed, the prevalence of polymorphic code is high, and researchers have noted that “94% of malicious executables are polymorphic” [5]. Compared to polymorphic malware, MM is even more complex, as it leverages numerous transformation techniques (successive and/or concurrent).

Researchers at Tripwire, among others, have posited that the rise in polymorphic or metamorphic malware is possibly tied to the current predominant signature-based security paradigm, wherein cyber threat intelligence-sharing has, to date, tended to be more heavily based upon the sharing of file hashes; hence, if the polymorphic or MM changes its file in each instance, potentially, “the effectiveness of defenses sharing threat intelligence about that piece of malware will drop drastically” [5].

III. Theoretical foundations and approach

The theoretical approach towards contending with metamorphic malware detection ranges from, by way of example, Ling et al., who focused upon leveraging NMF for detecting smaller subsets of the overarching set, via the utilization of structural entropy (which was deemed to exhibit greater promise than structural compression ratios) [6], to Begenholtz et al., who found that it is possible to accurately determine whether a file has been packed by a metamorphic packer “with an accuracy of up to 89.36% when trained on a single packer, even for samples packed by previously unseen packers” [7]; the latter study also focused upon leveraging Multilayered LSTM Networks for the involved detection. Along this vein, Ling et al. and many others have also focused upon structural entropy, such as via the use of Multilayer Perceptron (MLP) Neural Networks (NN) and other constructs, such as that of a Recurrent Neural Network (RNN) for behavioral feature extraction combined with a Feedforward Neural Network (FNN) (e.g., a Deep Feed Forward Concurrent Neural Network for classification, as put forth by Zhou [8]).

The premise is that when an executable file changes between states, such as from its native uncompressed state to a compressed, encrypted, etc. state, the file’s representative structural entropy also changes. According to Lyda et al., compressed or encrypted files have greater entropy values [9]. Leveraging this heuristic, Wojnowicz et al. utilized wavelet decomposition (which can successfully decompose complex information/patterns into lower rank representations) on the representative structural entropy of files to obtain the associated EWES, which provides insight into the potential use of packers, crypters, and protectors [10]. Moreover, while Singular Value Decomposition (SVD) has been widely used to obtain low-rank matrix approximation, the advantage of NMF when contending with structural entropy is that it is a fairly robust unsupervised learning approach for the analysis of high-dimensional data, and it can facilitate feature extraction from very large sparse matrices [11], whereas other approaches are not as readily able to process very large matrices due to various issues, including, but not limited to, missing entries or prolonged convergence [12]. The classic example involves a very large matrix A being factorized into, let us say, matrices B and C. Ultimately, the desire is that all the involved matrices have no negative elements [13]. However, if a prototypical method of matrix factorization (e.g., SVD) is used, the resulting SVD-based lower rank representation leads to both positive and negative elements (which is the antithesis of the intent to have no negative elements), thereby making interpretation quite challenging due to the ensuing ambiguity. In contrast to SVD, because NMF has the inherent constraint that the factorized matrices be comprised of non-negative (i.e., positive) elements, it can facilitate a more robust interpretation of the original matrix data, as it segues to a more intuitive structural representation by parts; as previously discussed in [12], the involved approximation/representation as the sum of positive elements (e.g., matrices, vectors, integers) is more intuitive, logical, and naturalistic given the matrices of positive integers. By leveraging the advantage of NMF's non-negative element constraint, various high-level features are more readily discerned from the hidden layers of the involved NN. Hence, the more naturalistic NMF-based approach reduces the need for feature engineering (i.e., a coarser and less elegant approach of extraction). Consequently, when the posited SOT is utilized, which starts at NMF and ends at a CWT, it is indeed possible to extrapolate upon the works of Ling et al., Begenholtz et al, and Wojnowicz et al., among others.

IV. Experimentation

A. Experimental Considerations

MM utilize various concealment/obfuscation methods while preserving the functionality of their intent. According to Borello et al., these methods can be classified as: data flow obfuscation (e.g., dead or junk code insertion, variable or register substition, instruction permutation or replacement, etc. [14]) and control flow obfuscation (e.g., code transposition, flattening — control flow flattening is a technique used not only to legitimately safeguard software from being reverse engineered, but is also illegitimately used by malware creators to obfuscate and hinder reverse engineering by cyber defenders, via the use of modification of the statement and loops in the code, layered obfuscation, etc. [15]). With regards to data flow obfuscation, Srdihara, et al. reported that by inserting a large amount of dead/junk code derived from benign files, the statistical properties of the ensuing MM morphed code could possibly be indistinguishable from benign codes [16]. With regards to control flow obfuscation, the transformed/obfuscated MM is semantically equivalent with regards to its original intent, but also immensely more difficult for detectors to analyze.

Various researchers have contributed to the detection of malware. For example, Ekhtoom et al. had classified MM families and obtained experimental results of 77% accuracy [17]. Bhattacharya et al. experimented with similarity measures and wavelet analysis and achieved an accuracy of 82.1% [18]. Bat-Erdene et al. experimented with entropy estimations and achieved an accuracy of 94.13% [19]. Alam et al. have asserted that they achieved a MM detection rate of 98.9% (with a false positive rate of 4.5%) [20].

B. Experimental Design & Implementation

Based upon the cited experimental consideration statistics, any posited MM detection should be in a similar range of detection efficacy to be of meaningful value-added proposition in the applied realm. This work chose to build upon the work of Ling et al., Begenholtz et al, and Wojnowicz et al., among others. An RNN paradigm was used for the behavioral feature extraction of the MM, and a FNN was utilized for the classification. However, one of the main contributions of this paper resided in the fact that an RCR-LSTM DLNN was utilized to support the RNN to FNN progression by facilitating a M2ED/output between the RNN and FNN; this would lead to improved discernment accuracy.

The RCR-LSTM DLNN accomplished its facilitation by operationalizing the posited SOT. The involved SOT in the experimentation for this paper progresses from NMF to MMF to Corresponding Wavelet Transform (CORWT) to an Enhanced CORWT (ECORWT), which was operationalized by way of a CWT PyWavelet Schema. The central aim of this approach was to arrive at a CWT paradigm, which does not substantively experience the energy leakage issues experienced by other commonly utilized transforms, such as Discrete Wavelet Transforms (DWT). For the involved experimentation, a particular NI of CWTs was utilized, via the referenced RCR-LSTM DLNN.

To successfully progress through the SOT, a non-conventional NMF approach is needed in the form of an Input Synthesis Model (ISM), which facilitates the MMF (the chosen method for ascertaining the involved multiscale structure and the delineation of the involved wavelets for a multi-resolution representation) [21] as well as, in turn, the determination of the MMF's CORWT, ECORWT, and the ensuing CWT. There is also a subtlety; certain operations are needed to fully transform the interim Gaussian

Composite Model (GCM) to a fully formed ISM, which then segues to the MMF. There is yet another subtlety. As illuminated in [12], the leveraging of a CWT PyWavelet schema (a Python-based open-source WT library) must be accompanied by the cognizance of the contained Mother Wavelets (i.e., families of Wavelets, which encompass both DWT and CWT); within each of these Wavelet families, there may be varied subordinate Wavelet subcategories, which are, generally speaking, differentiated by the number of coefficients (i.e., the number of vanishing moments, which refers to the state wherein the Wavelet coefficients are zero for those polynomials with a degree of at most $p-1$, and the scaling function alone can be utilized to represent the function) as well as the level of decomposition — as the number of vanishing moments increases, the polynomial degree of the wavelet also increases, the involved graph tends to become smoother, and it also turns out that the leveraging of CWT well enables the intricate structural characteristics of the NMF input, within the transform space, to be more amenable to the process of analysis and discernment [22][23]. The experimental design and implementation is summarized in Fig. 1 below.

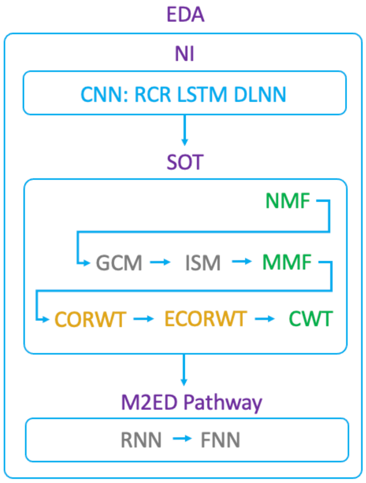

Figure 1. EDA Instantiation via an NI-enabled SOT for a M2ED Pathway

Ultimately, the experimental implementation involved three facets: utilization of an RNN for behavior feature extraction of the MM, utilization of a FNN for classification of the MM, and an RCR-LSTM DLNN-based NI to operationalize the SOT. The first facet addressed the static features (e.g., operation codes or opcodes, byte-level n-grams [extracted from, by way of example, Portable Executables or PEs], as a non-signature- based approach for detection, etc.) and dynamic features (e.g., recorded API calls) of the MM file. The feature vectors derived from the static and dynamic information were concatenated. The RCR-LSTM DLNN assisted the RNN in the conversion to a M2ED/output, which the FNN then utilized for classification; in other words, the RCR-LSTM DLNN facilitated the RNN to FNN progression.

C. *Experimental Results*

It should be noted that the utilized RNN was utilized for behavioral feature extraction. It should further be noted that the FNN was utilized for MM sample classification. While prototypical NNs have numerous layers, DLNN is a type of NN that is comprised of numerous hidden layers. Medina et al. had shown that the use of CNNs reduces the false positive rate [24]. Along this vein, Moradi et al. and others have described how the use of LSTMs addresses the gradient vanishing issue (a consequence of the derivative of the activation function used to instantiate the NN, which can be, by way of example, obviated by using an activation function, such as Rectified Linear Unit or ReLU instead of sigmoid), which besets RNNs [25]. The bespoke RCR-LSTM DLNN is one such CNN leveraging a LSTM. The RCR-LSTM DLNN was evaluated against other prototypical ML and DLNN methods. As just one indicator, the bespoke RCR-LSTM DLNN classification results are shown in Table 1 below. As a summary, a preliminary version of the posited bespoke RCR-LSTM DLNN method was able to achieve comparable ACC as other well-known methods, such as KNN, RNN, and RBF SVM; however, despite the fact that the posited bespoke method did not achieve the 98.9% rate (with a false positive rate of 4.5%) reported by Alam et al., it is hoped that future versions of the RCR-LSTM DLNN may possibly break the glass ceiling that is currently constraining the aforementioned methods. Future work will tell.

TABLE I. CLASSIFICATION RESULTS OF VARIOUS ML METHODS

Methods	Models	Accuracy (ACC)
Prototypical ML methods	Decision Tree (DT) [26]	82.4%
	Hidden Markov Models (HMM) [27]	87.3%
	Random Forest (RF) [28]	91.43%
	Sigmoid Support Vector Machine (SVM) [26]	95%
	k-Nearest Neighbor (KNN) [29]	97.6%
	Radial Basis Function (RBF) SVM [26]	97.9%
Prototypical DLNN methods	CNN [30]	96.96%
	RNN [31]	97.8%
Posited bespoke RCR-LSTM DLNN method	RCR-LSTM DLNN	97.9%

MM samples were obtained by using krmaxwell/maltrieve and jstrosch/malware-samples. The Cuckoo Sandbox was utilized to record the API calls and analyze the MM; however, while the use of API calls to unveil behavioral patterns was utilized to great effect by Hansen et al., Daeef et al., and others [32][33], it seems to have limited efficacy against potent MM. Prototypical ML libraries (e.g., Keras, Scikit-learn, etc.) were utilized. Experimental variations included PyTorch (PT), Tensorflow (TF), Caffe (CE), Caffe2 (CE2), and SciPy (SP). PT and TF were the favored implementations due to their prevalence and robust documentation. The choice of leveraging NMF, via the utilization of structural entropy (rather than structural compression ratios) seems to have been affirmed; after all, NMF's non-negative element constraint provides more ready discernment of high-level features from hidden layers. Likewise, the use of LSTM for the detection (to mitigate against the RNN deficiency of the gradient vanishing issue), seems to have been prudent; in addition, although Begenholtz et al., had experimented with multi-layer LSTM models, Catak et al. and others found that single-layer and multi-layer LSTM models attained similar classification outcomes [26]. For the experiment discussed in this paper, a single-layer LSTM was utilized, per Catak's findings.

On the entropy front, generally speaking, entropy refers to the measure of uncertainty pertaining to the data of an involved file, and the measurement value ranges between 0 to 8. The lower the value, the lower the probability that the code has been obfuscated, encrypted, etc. The higher the value, the higher the probability that the code has been obfuscated, encrypted, etc. [9]. A M2ED/output (associated with an enhanced wavelet decomposition) will segue to more accurate values for the involved measurement values. As an additional heuristic, MM coefficient scores tend to condense close to 1.0, whereas the substantive portion of benign files tend to have "smaller similarity coefficient scores as they are relatively far from 1.0" [34]. Preliminary experimentation has shown that the posited bespoke RCR-LSTM DLNN method does segue to enhanced measurement values, and the determination of entropy values is enhanced (i.e., M2ED), thereby providing greater confidence in utilizing the heuristic of "compressed or encrypted files have greater entropy values" [9].

A prototypical confusion matrix (utilized to depict the classification performance) was utilized for evaluation. True Positive Rate (TPR) equates to True Positive (TP)/Positive (P), wherein P = TP + False Negative (FN). Along this vein, False Positive Rate (FPR) equates to False Positive (FP)/Negative (N), wherein N= FP + True Negative (TN). Furthermore, Accuracy Rate (AR) equates to (TP + TN)/(P + N). The Receiver Operating Characteristic (ROC) curve was utilized to depict the classification performance at various classification thresholds with the two parameters of TPR and FPR. Following the lead of Zhou and others [8], Area Under the [ROC] Curve (AUC) was utilized for the measure of separability (i.e., classification performance). The performance of the nine classifiers cited in Table 1 was also supported by the utilization of N-fold cross-validation, via Waikato Environment for Knowledge Analysis (WEKA). As our posited approach is predicated upon an Adaptive Weighting System (AWS) [35], the subtle intent of cross-validation is somewhat obviated. For example, if all data samples were utilized to train the involved NN, the ensuing weights and bias values would tend to overfit (thereby setting the stage for poor performance again new, previously unseen data). To mitigate again overfitting, the convention is to separate the data into training data (e.g., 80%) and test data (e.g., 20%) so as to find an apropos balance. With an AWS, the mechanics of N-fold cross-validation become evidently more trite. The prototypical number of folds utilized is 10, and the involved experimentation uses this figure. The n-fold cross-validation provides a measure of quality (i.e., classification error) of each fold; axiomatically, the smaller the ensuing value, the better the performance. It was prudent to adhere to the standard of utilizing an artifically suppressed number of training iterations (a high number yields will result in higher performance) to provide a more realistic sense of performance. Generally, the performance at the first fold is better than that at the last fold. As noted in the next section, more experimentation will be conducted for augmenting the training data and studying classification performance [36].

V. Conclusion

The increase in cyber threat information feeds has provided an expanding corpus of malware samples to analyze. The discussed corpuses include, among others, krmaxwell/maltrieve and jstrosch/malware-samples. Meanwhile, as ML approaches have become more robust and sophisticated, ML-based MM detection approaches have improved as well. This is opportune due to a convergence of factors: (1) the required uptime and HA of ICS and DCS, among others, have increased, the necessity of IIOT sensors for an enhanced O&M CMP has also risen, (2) the necessity for higher resolution and greater reliability IIOT sensors, (3) the dependence upon APIs to detect for CMP-related issues, (4) the range of cyber-related vulnerabilities, particularly MM, which have plagued the APIs of IIOT sensors, (5) the dramatic rise and prevalence of MM, (6) the fact that strategic/critical infrastructure IIOT sensors and OT are part of the top five "currently manifesting risks," as noted by the WEF.

The theoretical approach utilized, to operationalize the discussed M2ED/output for an improved MM detection paradigm, involved non-conventional NMF and MMF (used in conjunction so as to facilitate the capture of the structure and content of the involved matrices so as to attain higher resolution and EDA) to more elegantly segue to CWT (via the intermediary steps of CORWT and ECORWT). This NMF-MMF-CWT paradigm, operationalized by the discussed RCR-LSTM DLNN (which supported the RNN to FNN progression), was the key SOT for EDA (i.e., M2ED) and one of the main contributions of this paper. The RCR LSTM DLNN amalgam brings several value-added propositions to bear: (1) the CNN amalgam construct itself reduces the false positive rate, (2) the RCR construct facilitates more robust bounds tightening, and (3) the LSTM mitigates against the RNN deficiency of the gradient vanishing issue. The operationalization of the SOT (the leveraging of wavelet decomposition on the representative structural entropy to ascertain the associated EWES) for an enhanced EWES, which was referenced as M2ED, provided a form of Indications and Warnings (I&W) for the potential use of packers, crypters, and protectors. Future work will involve more quantitative experimentation in this area.

Overall, the paper discussed the theoretical foundations and approaches utilized within this ecosystem, various experimental designs, and results related to MM detection. The paper also explored various pertinent techniques and methods, including leveraging RCR, LSTM, DLNN, NMF, MMF, CORWT, ECORWT, CWT, RNN, and FNN. The paper further details an experimental design using a bespoke RCR-LSTM DLNN method and presents the results, comparing them with other ML methods. The paper's focus on MM detection should be of relevance to the current cybersecurity landscape, particularly as attacks on OT for strategic/critical infrastructure operations are among the top currently manifesting risks.

Acknowledgment

This research is supported by the Decision Engineering Analysis Laboratory (DEAL), an Underwatch initiative of VTIRL, VT. This is part of an ongoing VTIRL technical series, on behalf of the Quality Assurance/Quality Control (QA/QC) unit, to advance the involved TRLs.

References

[1] A. Waldman, "Dragos: Ransomware topped ICS and OT threats in 2021," Tech Target, Feb 23, 2022, Accessed: July 28, 2023. [Online]. Available from: https://www.techtarget.com/searchsecurity/news/252513714/Dragos-Ransomware-topped-ICS-and-OT-threats.

[2] I. Bramson, "Vulnerable Today, Hacked Tomorrow: How a Lack of OT Cybersecurity Affects Critical Infrastructure," Cyber Defense Magazine, May 6, 2022, Accessed: July 28, 2023. [Online]. Available from: https://www.cyberdefensemagazine.com/vulnerable-today/

[3] World Economic Forum, Marsh McLennan, and Zurich Insurance Group, "Global Risks Report 2023," World Economic Forum, January 11, 2023, Accessed: July 28, 2023. [Online]. Available from: https://www.weforum.org/reports/global-risks-report-2023/in-full/1-global-risks-2023-today-s-crisis/.

[4] McKinsey & Company, "How to Enhance the Cybersecurity of Operational Technology Environments," March 23, 2023, Accessed: July 28, 2023. [Online]. Available from: https://www.mckinsey.com/capabilities/risk-and-resilience/our-insights/cybersecurity/how-to-enhance-the-cybersecurity-of-operational-technology-environments.

[5] SSL, "Polymorphic Malware and Metamorphic Malware: What You Need to Know," March 25, 2021, Accessed: July 28, 2023. [Online]. Available from: https://www.thesslstore.com/blog/polymorphic-malware-and-metamorphic-malware-what-you-need-to-know/.

[6] Y. Ling, N. Sani, and M. Abdullah, "Nonnegative Matrix Factorization and Metamorphic Malware Detection. J Comput Virol Hack Tech 15, 2019, pp. 195–208, doi: 10.1007/s11416-019-00331-0

[7] E. Bergenholtz, E. Casalicchio, D. Ilie, and A. Moss, "Detection of Metamorphic Malware Packers Using Multilayered LSTM Networks," Information and Communications Security. Lecture Notes in Computer Science vol 12282, Springer, Cham, doi: 10.1007/978-3-030-61078-4_3

[8] H. Zhou, "Malware Detection with Neural Network Using Combined Features," Communications in Computer and Information Science, vol 970, 2019, pp 96–106, doi: 10.1007/978-981-13-6621-5_8

[9] R. Lyda and J. Hamrock, J, "Using Entropy Analysis to Find Encrypted and Packed Malware," IEEE Secur. Priv. 5(2), 2007, pp. 40–45, doi: 10.1109/MSP.2007.48.

[10] M. Wojnowicz, G. Chisholm, M. Wolff, and X. Zhao, "Wavelet Decomposition of Software Entropy Reveals Symptoms of Malicious Code," J. Innov. Digit. Ecosyst. 3(2), 2016, pp. 130–140, doi: 10.48550/arXiv.1607.04950.

[11] N. Gillis, "The Why and How of Nonnegative Matrix Factorization," Regularization, Optimization, Kernels, and Support Vector Machines, Jan 2014, pp. 257-291, doi: 10.48550/arXiv.1401.5226.

[12] S. Chan, M. Krunz and B. Griffin, "Adaptive Time-Frequency Synthesis for Waveform Discernment in Wireless Communications," 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 2021, pp. 0988-0996, doi: 10.1109/IEMCON53756.2021.9623140.

[13] A. Zaeemzadeh, M. Joneidi, B. Shahrasbi, and N. Rahnavard, "Missing Spectrum-Data Recovery in Cognitive Radio Networks Using Piecewise Constant Nonnegative Matrix Factorization," MILCOM 2015 - 2015 IEEE Military Communications Conference, 2015, pp. 238-243, doi: 10.1109/MILCOM.2015.7357449.

[14] J. Borello and L. Mé, "Code Obfuscation Techniques for Metamorphic Viruses," J. Comput. Virol. 4(3), 2008, pp. 211–220, doi: 10.1007/s11416-008-0084-2.

[15] H. Xu, Y. Zhou, and J. Ming, "Layered Obfuscation: A Taxonomy of Software Obfuscation Techniques for Layered Security," Cybersecurity 3(1):9, 2020, pp. 1-18, doi: 10.1186/s42400-020-00049-3.

[16] S. Sridhara and M. Stamp, "Metamorphic Worm That Carries Its Own Morphing Engine," J Comput. Virol. Hacking Tech. 9(2), 2013, pp. 49-58, doi: 10.1007/s11416-012-0174-z.

[17] D. Ekhtoom, M. Al-Ayyoub, M. Al-Saleh, M. Alsmirat, and I. Hmeidi, "A Compression-Based Technique to Classify Metamorphic Malware," 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), 2016, pp. 1–6, doi: 10.1109/AICCSA.2016.794580.

[18] A. Bhattacharya and R. Goswami, "Data Mining Based Detection of Android Malware," Proceedings of the First International Conference on Intelligent Computing and Communication. Advances in Intelligent Systems and Computing, v 458, 2016, pp 187–194, doi: 10.1007/978-981-10-2035-3_20.

[19] M. Bat-Erdene, H. Park, H. Li, H. Lee, M. Choi, "Entropy Analysis to Classify Unknown Packing Algorithms for Malware Detection," Int J Inf Secur 16(3), 2017, pp. 227–248. doi: 10.1007/s10207-016-0330-4.

[20] S. Alam, I. Traore, and I. Sogukpinar, "Annotated Control Flow Graph for Metamorphic Malware Detection," The Computer Journal, vol. 58, no. 10, Oct. 2015, pp. 2608-2621, doi: 10.1093/comjnl/bxu148.

[21] R. Kondor, N. Teneva, and P. Mudrakarta, "Parallel MMF: A Multiresolution Approach to Matrix Computation," Arxiv, 2015, Accessed: July 28, 2023. [Online]. Available from: https://arxiv.org/abs/1507.04396.

[22] P. Addison, “Introduction to Redundancy Rules: The Continuous Wavelet Transform Comes of Age,” Philosophical Transaction of the Royal Society A., 2018, pp. 1-38, doi: https://doi.org/10.1098/rsta.2017.0258.

[23] A. Levinskis, “Convolution Neural Network Feature Reduction Using Wavelet Transform,” Electronics and Electrical Engineering, vol. 19, 2013, pp. 61-64, doi: 10.5755/j01.eee.19.3.3698.

[24] E. Medina, M. Petraglia, J. Gomes, and A. Petraglia, “Comparison of CNN and MLP classifiers for Algae Detection in Underwater Pipelines,” Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), 2017, pp. 1-6, doi: 10.1109/IPTA.2017.8310098.

[25] M. Mahvash, S. Sadrossadat, and V. Derhamit, “Long Short-Term Memory Neural Networks for Modeling Nonlinear Electronic Components,” IEEE Transactions on Components, vol 11 Issue 5, doi: 10.1109/TCPMT.2021.3071351.

[26] C. Ferhat, Y. Ahmet, E. Ogerta, and A Javed, “Deep Learning Based Sequential Model for Malware Analysis using Windows exe API calls,” PeerJ Comput Sci vol 6, 2020, doi: 10.7717/peerj-cs.285.

[27] C. Annachhatre, T. Austin, and M. Stamp, “Hidden Markov Models for Malware Classification,” J. Comput. Virol. Hack. Tech. 11(2), 2014, pp. 59–73, doi: 10.1007/s11416-014-0215-x.

[28] B. Khamma, “Ransomware Detection Using Random Forest Technique,” ICT Express, Vol 6, Issue 4, December 2020, pp. 325-331, doi: 10.1016/j.icte.2020.11.001.

[29] G. Dahl, J. Stokes, and L. Deng, “Large-scale Malware Classification Using Random Projections and Neural Networks,” IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013, pp. 3422-3426, doi: 10.1109/ICASSP.2013.6638293.

[30] S. Lad and A. Adamuthe, “Malware Classification with Improved Convolutional Neural Network Model,” I.J. Computer Network and Information Security, 2020 pp. 30-43. doi: 10.5815/ijcnis.2020.06.03.

[31] H. Madani, N. Ouerdi, A. Boumesaoud, and A. Azizi, “Classification of Ransomware Using Different Types of Neural Networks,” Sci Rep. 12(1), March 19, 2022, pp. 4770. doi: 10.1038/s41598-022-08504-6.

[32] S. Hansen, T. Larsen, M. Stevanovic, and J. Pedersen, “An Approach for Detection and Family Classification of Malware Based on Behavioral Analysis,” Proceedings of the 2016 International Conference on Computing, Networking, and Communications (ICNC), 2016, pp. 1-5, doi: 10.1109/ICCNC.2016.7440587.

[33] A. Daeef and A. Al-Najii, “Features Engineering for Malware Family Classification Based API Call,” Computers 11(11), 2022, doi: 10.3390/computers11110160.

[34] L. Yeong, N. Sani, M. Abdullah, N. Hamid, “Nonnegative Matrix Factorization and Metamorphic Malware Detection,” Journal of Computer Virology and Hacking Techniques 15, 2019, pp. 195-208, doi: 10.1007/s11416-019-00331-0.

[35] S. Chan and P. Nopphawan, “Accelerant Facilitation for an Adaptive Weighting-Based Multi-Index Assessment of Cyber Physical Power Systems,” 2023 IEEE 3rd International Conference in Power Engineering Applications (ICPEA), 2023, pp. 156-162, doi: 10.1109/ICPEA56918.2023.10093212.

[36] K. O. Babaagba, Z. Tan and E. Hart, “Improving Classification of Metamorphic Malware by Augmenting Training Data with a Diverse Set of Evolved Mutant Samples,” 2020 IEEE Congress on Evolutionary Computation (CEC), 2020, pp. 1-7, doi: 10.1109/CEC48606.2020.9185668.