2018 10th International Conference on Cyber Conflict
CyCon X: Maximising Effects
T. Minárik, R. Jakschis, L. Lindström (Eds.)
2018 © NATO CCD COE Publications, Tallinn

Neural Network and Blockchain Based Technique for Cyber Threat Intelligence and Situational Awareness

Roman Graf
Austrian Institute of Technology GmbH
Vienna, Austria
roman.graf@ait.ac.at

Ross King
Austrian Institute of Technology GmbH
Vienna, Austria
ross.king@ait.ac.at

Abstract: Protecting Critical Infrastructure (CI) against increasing cyber threats has become as crucial as it is complicated. To be effective in identifying and defeating cyber attacks, cyber analysts require novel distributed detection and reaction methodologies based on information security techniques that can automatically analyse incident reports and securely share analysis results between Critical Infrastructure stakeholders. Our goal is to provide solutions in real-time that could replace human input for cyber incident analysis tasks (triage) to classify cyber incident reports, find related reports in a fast and scalable way, eliminate irrelevant information, and automate reporting life-cycle management. Our effective and fast incident management method is based on artificial intelligence and can support cyber analysts in establishing cyber situational awareness, and allow them to quickly adopt suitable countermeasures in the case of an attack. In this paper, we evaluate deep autoencoder neural network supported by Blockchain technology as a system for incident classification and management, and assess its accuracy and performance. This approach should reduce the number of manual operations and save storage space. We used a Blockchain smart contract technique to provide an automated trusted system for incident management workflow that allows automatic acquisition, classification and enrichment of incident data. We demonstrate how the presented techniques can be applied to support incident handling tasks performed by security operation centres.

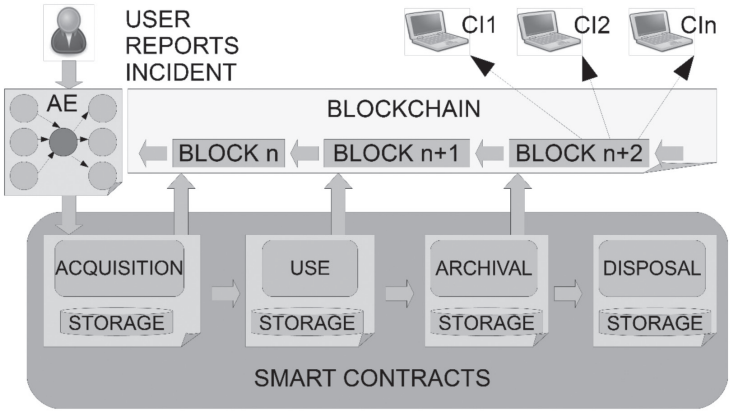
Keywords: *cyber threat intelligence, neural network, blockchain*

1. INTRODUCTION

Cyber Situational Awareness (SA) [1] is a perception of security and threat situations coupled with current and future impact assessment. In recent years, researchers in the SA field have created increasingly complex tools across many application domains. Speed of events, data overload, and meaning underload [2] make real-time SA of cyber operations very difficult to evaluate. Addressing data that is often vague and imprecise, we have to rely on imperfect information to detect real attacks and to prevent an attack from happening through appropriate risk management. Security Operation Centre (SOC) analysts receive a huge amount of daily threat reports. These analysts face challenges finding relevant information in large, complex data sets when exploring data to discover patterns and insights and following organisation business processes, such as proper acquisition, use, archiving and disposal of threat reports. For humans to be effective in identifying and defeating cyber attacks, novel tools that can fill the gap between cyber data and situation comprehension are highly desired. The research presented here is designed to aid in developing a system (see Figure 1) that will automatically support a cyber analyst by analysing and classifying incoming cyber incidents by searching similar high severity cyber incidents that could affect cyber SA, and by life-cycle management of the incident.

Analysis is triggered by a cyber incident report generated by one of the stakeholders in the CI network. The incident analysis can be performed for large amounts of data by using a solid knowledge base (KB), and employing one of the available incident analysis tools. A deep autoencoder (AE) method can be used to analyse existing KB or particular large dataset. The primary purpose of designing a deep autoencoder for SA is to increase the speed of sharing highly severe information and to enable fast and trustworthy cyber incident classification, without the need for substantial human involvement. In our study, we compare existing cyber threat intelligence tools and techniques, describe automatic cyber intelligence analysis approach using a deep autoencoder neural network, and present evaluation results. We leverage expertise collected in available cyber intelligence tools with the power of the neural networks approach.

FIGURE 1. THE OVERVIEW OF ESTABLISHING THE CYBER SITUATIONAL AWARENESS USING NEURAL NETWORKS (AE) AND SMART CONTRACTS FOR INFORMATION CLASSIFICATION AND LIFE-CYCLE MANAGEMENT.



The primary contribution of this work is a real-time solution that could replace human input for a huge number of cyber incident analysis tasks. Another is a methodology, developed to improve information organization and access in cyber security information systems based on automatic classification of cyber security documents according to their expected threat level. We hypothesise that the application of Smart Contracts based on the existing Blockchain technology Ethereum [3] can solve some SA problems. The main purpose of designing Smart Contracts for SA is to enable rapid and trusted cyber incident classification and management, without the need for a large centralised authority. We propose that Smart Contracts based on decentralised assets such as Ethereum can reduce effort for incident life-cycle management and manual analysis costs. Novel techniques that can automatically make predefined decisions obvious by using Smart Contracts can help identify and defeat cyber attacks.

In our context, a Smart Contract basically is a piece of software that fixes and verifies negotiated behaviour and cannot be manipulated because it is distributed and executed on multiple nodes on a Blockchain. Another value of using Smart Contracts is that once deployed, it can function automatically, without the need for human interaction. In our proposed threat intelligence analysis system, we describe the incident handling procedure and instructions using a Smart Contract programming language (Solidity) and upload this Smart Contract to a Blockchain instance (a private Ethereum network). The source code of the Smart Contract defines instructions and rules; for our system, we created 'Acquisition', 'Use', 'Archival' and 'Disposal' Smart Contracts (see Figure 1). The state of the Smart Contracts is stored on the Blockchain and is transparent and accessible to all registered community members. The Smart Contract code is executed in parallel by a network of miners under consensus regarding the outcome of the

execution. The execution of the Smart Contract results in an update of the contract's state (BLOCKn+2) on the Blockchain that is synchronised with every participating user (CI1-CIn) through standard peer-to-peer mechanisms and a Proof-of-Work-based consensus mechanism. An incident report produced by one of the users (security expert protecting CIs) goes through the Smart Contracts and is handled automatically, according to the programmed instructions.

The management system is aimed at the automatic management of threat reports provided by threat analysis tools such as CAESAIR,[1] IntelMQ[2], or MISP[3] and should provide effective decision support for a SOC operator. Compared to manual classification, automatic classification by threat level can significantly support and accelerate reaction time of an SOC analyst. For example, the Collaborative Analysis Engine for the Situational Awareness and Incident Response (CAESAIR) tool [4] supports various security information correlation techniques and provides customizable import capabilities from a multitude of security-relevant sources. These sources include a custom repository, open source intelligence (OSINT) feeds and IT-security bulletins, as well as a standardised vulnerability library (Common Vulnerabilities and Exposures – CVE). CVEs are especially important for Smart Contracts with regard to likelihood assessments based on game theory [5] that implements risk scoring [6]. Employing CAESAIR with CVE scoring [7] and extending it by automated tagging can provide valuable input for information classification and life-cycle management. Such a system can be implemented using Smart Contracts created for a particular organization. Each institution may have multiple classification profile definitions dependent on the network, CI and the role of the cyber analyst.

This paper is structured as follows. Section 2 gives an overview of related work and concepts. Section 3 explains the cyber incident classification workflow. The cyber incident life cycle issues are covered in Section 4, Section 5 presents the experimental setup, applied methods end evaluation and Section 6 concludes the paper.

2. RELATED WORK

Threat intelligence in the cyber security (CS) realm is provided by a number of cyber incident analysis tools. For example, the CAESAIR tool provides analytical support for security experts carrying out cyber incident handling tasks on national and international levels, and facilitates the identification of implicit relations between available pieces of information. IntelMQ is an open source tool collaboratively developed by Austrian CERT and other parties aiming at parsing and correlating cyber incidents. MISP, the Malware Information Sharing Platform is another open source tool that performs automatic data correlation by finding relationships between

1 http://caesair.ait.ac.at
2 https://github.com/certtools/intelmq
3 https://github.com/MISP/MISP

attributes and indicators from malware, attack campaigns, or analysis. It incorporates an indicator database to store technical and non-technical information about malware samples, incidents, attackers and intelligence; and a sharing functionality to facilitate data exchange using different models of distribution.

The autoencoder approach is widely used for different analytical tasks. A machine learning framework based on recursive autoencoders [8] can be used for sentence-level prediction of sentiment label distributions. A very deep autoencoder [9] is employed for content-based image retrieval. In our approach, we are using this method for similarity searches. The advantage of the autoencoder method is that it learns automatically from examples. The autoencoder makes use of neural networks which are already in use by latent semantic analysis for text categorization [10] to reduce dimensionality and to improve performance. Another application [11] employs an artificial neural network to improve text classifier scalability. Classification methods implemented in the previously mentioned threat intelligence tools suffer from large vector sizes and are less effective as the number of incidents rise. The main drawback of existing text classification methods, such as SVM [12], Word Embeddings Neural Networks or the Gensim tool is that they require a huge database for training to provide meaningful results, but expected SOCs datasets are not large enough for such semantic-based tasks. Another common disadvantage of these techniques is the lack of results transparency due to employing vectors containing real-valued numbers. These tools provide results, but it is difficult to explain how the results were calculated. In particular, the SVM approach is limited by the choice of the kernel. Another disadvantage is the inability to handle unknown words or words which were not included previously in the training vocabulary. Consequently, for the particular use case of threat incident classification task for SOCs, we suggest using the autoencoder solution that scales well because of the small vector size while maintaining a high level of accuracy.

Multiple researchers are developing an automated technology that will support an information classification system. An attempt to classify the relationships between documents and concepts [13] employs principles of ontology. To improve information organization and access in construction management, a methodology [14] was developed based on automatic hierarchical classification of construction project documents according to project components. A survey of various cyber attacks and their classification [15] attempted to develop an ontology for cyber security incidents. They classify by characteristics, and by purpose and motivations. Additionally, cyber attacks can be classified based on the severity of involvement, scope, or network types with multiple sub classification terms. Contrary to this approach, we classify only by threat level that can differ from organisation to organisation. Our goal is to focus human expert resources on the most urgent incidents important for a particular organisation. An information life-cycle model described in [16] is also applicable to

the CS domain. Cyber incident reports are acquired, analysed and become outdated. Effective automatic classification, retention and disposal policies can mitigate risks to data and make information management more effective. Classification of data enables a company or SOC to focus their resources toward the most valuable or urgent incidents and to handle less valuable incidents, automatically saving time and other costs.

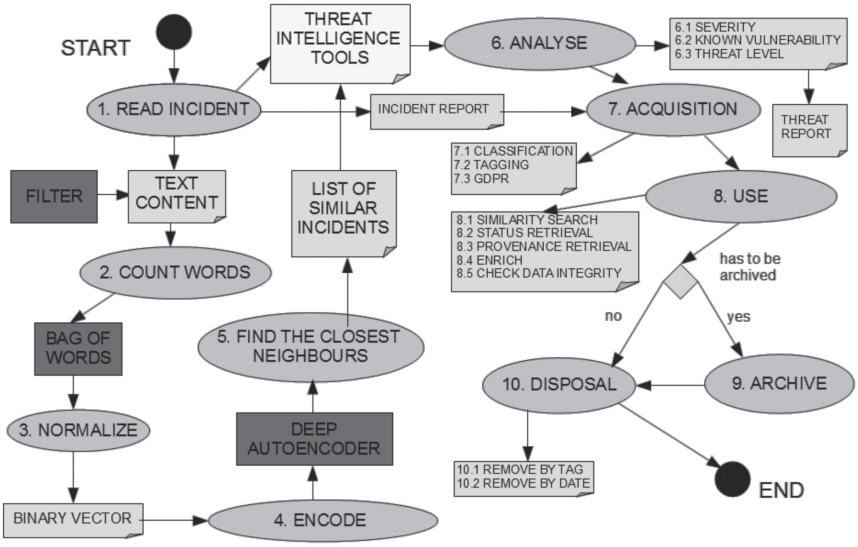
Because members of a CI network do not necessarily trust each other, do not have a central authority and have a need to store and share the life-cycle state of the incident, we suggest a Blockchain-based solution for life-cycle management. An overview of the Blockchain technology and its potential to facilitate money transactions, Smart Contracts design, automated banking ledgers and digital assets is provided in [17]. A Blockchain platform comparison [18] discusses five general-use Blockchain platforms and looks at how Blockchain technology can be used in applications outside of Bitcoin to build custom applications on top of it. This comparison suggests that Ethereum is currently the most suitable and well-established platform. Therefore, for cyber incident analysis we employ an Ethereum Blockchain (specifically, the Pyethereum implementation), which supports a focused Smart Contracts testing environment without the need of mining. In the proposed system, we intend to apply Smart Contracts for cyber incident classification and life-cycle management, which is unique for the given domain.

3. CYBER INCIDENT CLASSIFICATION USING AUTOENCODER

For our study, we assume that a cyber expert is responsible for a CI and detects suspicious behaviour in the system. The expert needs more information to select the correct mitigation strategy. She must collect and analyse all the available information related to ongoing and previous attacks for the particular use case, and transform it into actionable intelligence. Security information such as incident reports, vulnerability alerts, advisories, bulletins etc., usually come in the form of semi-structured text documents. Acquiring cyber threat intelligence from such documents requires manually reviewing and discerning what significant information they can find, and identifying implicit correlations among them in order to estimate their impact and outline possible mitigation strategies. To avoid this manual effort, the CIs expert can provide an incident report as an input to a deep autoencoder and receive a threat report back if it has sufficient severity. An automatic approach delivers a significant improvement in terms of personnel costs when compared to manual cyber incident handling. As a result, an analyst has the up-to-date SA status and we ensure fast and scalable information exchange and enrichment.

The idea behind applying the autoencoder approach is that we can map N-dimensional data onto the M orthogonal directions in which the data have the most variance and form a lower dimensional subspace. The acceptable drawback of this conversion is that in the remaining orthogonal directions we lose information about the original data point location.

FIGURE 2. THE WORKFLOW FOR CLASSIFICATION AND LIFE-CYCLE MANAGEMENT OF CYBER INCIDENT USING AUTOENCODER AND SMART CONTRACTS.



We employ a deep autoencoder that was trained as described in the workflow shown in Figure 2. The workflow execution starts with reading the incident report (1) and parsing the report content. Input data along with the expert profile settings, which are specific to the organisation, are converted to a binary vector using the 'bag of words' technique (2) and after the normalization step (3) passed to the autoencoder in encoded form (4). In this step, we compile the words most used in documents. The remaining vector is comprised of word counts irrespective of order. For simplicity, we use a binary count where we mark 1 if a word count is bigger than 0, and 0 if the given word is not present in an original document. Additionally, we ignore stop words (words with no discriminatory power, such as common articles and prepositions, that we do not need in analysis). To achieve reasonable performance and scalability, we reduce each vector to a much smaller vector that still comprises enough information about the content of the document. In the next step, we train the neural network to reproduce its input vector as its output. This forces it to compress as much information as possible into the 10 numbers in the central bottleneck. These 10 numbers are then a result of deep autoencoder training and a good way to compare documents (5) in a fast

and scalable way using the cosine similarity method. In the next step, we merge the detected related incidents with institutional settings and decide which priority level (see Equation 1) should be applied to the given incident. The compressed vectors are stored on the hidden level of neural network (see Table 1).

$$P = f(I_r, W_r, W_o, T_s, W_s) \qquad (1)$$

Equation 1 shows the incident priority level P that returns the value – either 0 that corresponds to 'Low' or 1 representing 'High'. Priority level is a function of aggregated incident evaluation metrics, which depend on basis indicators, such as 'number of related incidents' I_r, 'number of related words' W_r, 'number of original words' W_o, 'detected significant terms' T_s and 'vulnerability score' V_s.

4. CYBER INCIDENT MANAGEMENT USING SMART CONTRACTS

We evaluate the application of Smart Contracts to classify and manage incident reports labelled by the autoencoder as a high priority threat. Smart Contracts can be used to estimate that the reported cyber incident is of high relevance, to remove it after some predefined time, to tag it by acquisition, to search by tag, to assign access rights (confidential, private, sensitive, public), to periodically check data integrity (preventing manual or hardware corruption), or to determine data provenance. Our goal is to save storage space, improve performance and to keep information up-to-date in a trustworthy way by leveraging the distributed nature of Blockchain technology. Once a Smart Contract is triggered, the analysis result is automatically propagated among all participants through inherent Blockchain mechanisms. One of the advantages of this approach is that Smart Contracts cannot be changed or compromised without being detected (through hashed transactions) and that the messages can be verified to originate from a trusted source (through public key encryption). After incident acquisition, a Smart Contract performs the classification of a report by threat level, stores the obtained threat level on a Blockchain and initiates the life-cycle management process for the given incident. In the next step, this report will be used, archived and disposed.

We employ four Smart Contracts for cyber incident processing, as depicted in Figure 2. The workflow execution after the classification steps performed by the autoencoder proceeds with the analysis of an incident report by reading and parsing the report content enriched with the classification results (6). Input data, along with organization-specific expert profile settings, are passed to the first Smart Contract 'acquisition' (7), which employs one of the threat intelligence tools. Classification occurs by employing

incident text, split by words or phrases, specific terms separated by low, middle and high threat relevance. We compute risk points, counting how many of terms are included in the incident report for each threat level. For threat level calculation, we either estimate threat level by applying thresholds for each level or we employ the weighted method from Formula 2, where we additionally multiply the calculated points on each threat level with a constant which represents the weight of the related threat level. The threat level scale ranges from 1 to 3, where 1 is 'low threat' and 3 is 'high threat'. Risk points RP is a sum of high risk points H_{rp} multiplied by high threat weight HT_w, middle risk points M_{rp} multiplied by middle threat weight MT_w and low risk points L_{rp} multiplied by low threat weight LT_w.

$$RP = H_{rp} * HT_w + M_{rp} * MT_w + L_{rp} * LT_w \quad (2) \qquad T_l = \begin{cases} 3(high) & if\ RP > HT_t, \\ 2(middle) & if\ RP \geq MT_t, \\ 3(low) & else\ RP < MT_t, \end{cases} \quad (3)$$

Where HT_w=3, MT_w=2, LT_w=1 and HT_t=10, MT_t=3. Threat level T_1 can be inferred using high threat HT_t and middle threat MT_t thresholds and weighted risk points RP from Formulas 2 and 3. The acquisition step (7) is split into different tasks. Automatic classification by threat level defines one of three threat levels: 'high' level requires fast reaction and mediation steps, triage process; 'medium' level assumes detection of 'Indicator of Corruption' (IoC) or metrics that indicate possible vulnerabilities, and requires SW update; and 'low' level addresses regular cyber security information and logs, and requires attention but should not necessary be a threat. Tagging means that specific tags can be assigned to a report to make it easier to find, shift or remove later. Removing personal information from the incident report to protect personal data may be required (by the European GDPR) before storing a normalised version of the incident. In the 'using' step (8), the workflow supports an automated similarity search, status and provenance retrieval, and enrichment with data and metadata periodic check for data integrity (using the hash of the incident report). Finally, depending on the threat level after some period of time, the incident can be archived (step 9) or removed e.g. by date or by tag (step 10).

We believe that this automatic smart-contracts-based approach would substantially support incident classification and management and could be used by analysts for the defence of CI. The suggested method would make SA analysis less cost-intensive and would perform with higher throughput. However, as is typical in this area, a human-based approach performs with higher accuracy.

5. EXPERIMENTAL EVALUATION

In the evaluation section, we measure how accurate our automated computations are and how long it takes for the deep autoencoder to make its calculations. Additionally, we report on measurements of the automated cyber incident classification and how long it takes for Smart Contracts to be executed and validated. We carried out measurements for several incident reports. The goal of this evaluation was to leverage the domain expert knowledge base for cyber incident classification and management as described in the workflow (see Figure 2), pointing out threat level relevant for SA.

A. Evaluation Data Set

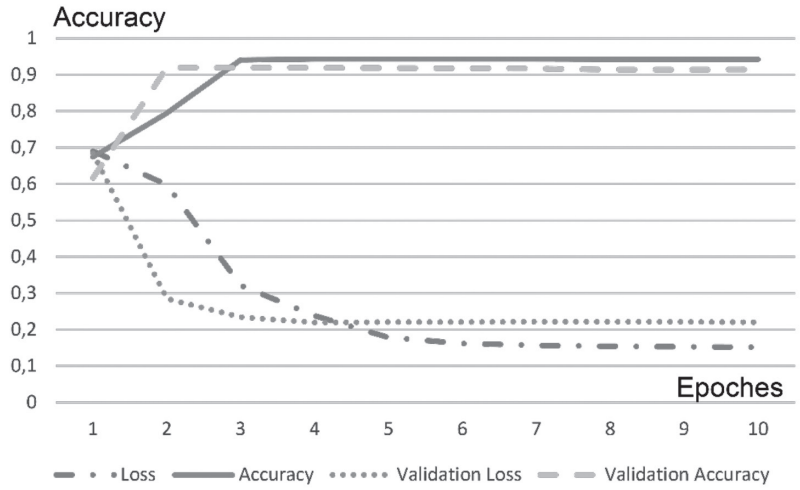
The cyber analyst's goal is to prioritise a detected cyber incident, either to mitigate it or to perform some other cyber incident response. For this test, we assumed that our CI is a financial organisation that employs MS Office products on Windows OS and using software products such as Internet Explorer, Firefox, Adobe, etc. The dataset used was aggregated from OSINT sources on the Internet. The dataset contained 5,850 training documents and 584 test documents. We evaluated cyber incident reports from the 'seclists' feed[4] from the last three years addressing four report categories. The 'fulldisclosure' category contained messages from the public, a vendor-neutral forum for detailed discussion of vulnerabilities and exploitation techniques, as well as tools, papers, news, and events of interest to the community. The 'bugtraq' category is a general security mailing list. The 'pen-test' category discloses techniques and strategies that would be useful to anyone with a practical interest in security and network auditing. The 'nmap-dev' category comprises an unmoderated technical development forum for debating ideas, patches, and suggestions regarding proposed changes to Nmap[5] and related projects. The specific cyber security terms were obtained from the CS glossary.[6] We anticipated that employing the described autoencoder and Smart Contracts approach should classify cyber incidents among a very large number of incident reports facilitating further cyber analysis and incident management.

B. Experimental Results and Interpretation

This evaluation took place on an Intel Core i7-3520M 2.66GHz computer using Python on Ubuntu OS. We performed a total of 10 training iterations (epochs) for the autoencoder. The autoencoder training and accuracy calculation process took about 262 seconds (see Figure 3). This figure shows that loss and validation loss decreased and accuracy and validation accuracy increased with each epoch. A final accuracy of 0.942 was achieved; this demonstrates how well input is reconstructed compared to the output.

4 http://seclists.org/
5 https://nmap.org/
6 https://scottschober.com/glossary-of-cybersecurity-terms/

FIGURE 3. ACCURACY AND LOSS CHARACTERISTICS BY AUTOENCODER TRAINING.



The neural network used a total of 502,000 parameters during the autoencoder training. The summary of the neural network training is presented in the Table 1. The neural network is composed of 1 input layer and 5 hidden layers. The number of neurons in these layers range from 10 to 2,000. Most layers use a rectified linear unit (ReLU) as an activation function. The last decoding layer employs a sigmoid activation function.

TABLE 1. SUMMARY OF THE DEEP AUTOENCODER TRAINING PROCESS.

Layer	Type	Activation Function	Neurons #	Parameters #
Input layer	InputLayer	ReLU	2,000	0
Hidden layer 1	Dense	ReLU	2,000	4,002,000
Hidden layer 2	Dense	ReLU	250	500,250
Hidden layer 3	Dense	ReLU	10	2,510
Hidden layer 4	Dense	ReLU	250	2,750
Hidden layer 5	Dense	Sigmoid	2,000	502,000

The autoencoder model simply maps an input to its reconstruction. To achieve this, we first train an autoencoder until it reaches the stable train/validation loss value. The deep autoencoder system starts the SA analysis with incident content retrieval, which is converted to an input vector by using word counts. This input vector then goes through encoding in multiple hidden layers and is reconstructed to an output layer after decoding in the final layers. Having trained the model, we were able to retrieve

the middle layer of the autoencoder model with the smallest number of neurons (10). Therefore, we retrieved trained 10-number-long IDs for each of the 584 test vectors and iterated this over all of the document vectors (10-numbers-long each) calculating a cosine similarity value for each document. For instance, the trained vector of the query incident report 'bugtraq-2017-Aug-1.txt' containing 10 numbers is [-8.73114914e-10, 1.01575899e+01, 2.12457962e-09, 1.29858088e+00, 2.67755240e-09, 9.32977295e+00, 4.54857439e-01, -5.82076609e-11, 8.55403137e+00, 5.52972779e-09]. This vector can be used for fast and scalable similarity search. Computation demonstrated that, for the given incident report, the first three most similar documents are: 'nmap-dev-2017-q2-8.txt, fulldisclosure-2017-Jan-68.txt, fulldisclosure-2015-Oct-71.txt'. During the correlation calculation using the deep autoencoder, there was a minor fluctuation of accuracy value in the last epochs (between 0.942 and 0.943). This is because the autoencoder employs a restricted Boltzmann machine (RBM), which treats the word counts as probabilities and makes use of random values in calculations. Therefore, it is possible that the highest level of accuracy can be achieved before all of the epochs are calculated (epoch 4 in our case).

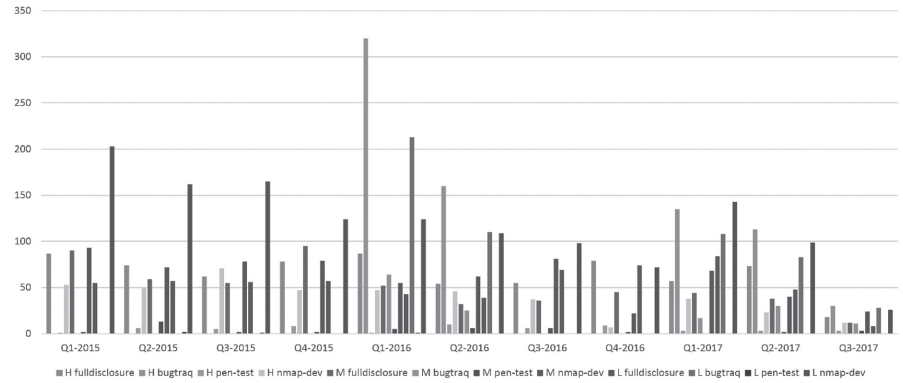
TABLE 2. EXCERPT OF CLASSIFICATION RESULTS FOR CYBER INCIDENT REPORTS BY THEIR ACQUISITION USING SMART CONTRACTS.

Incident ID	Related Incident ID	Similarity	Source	Block-chain ID	Time (sec)	Terms #	Threat Level
Fulldisclosure-2017-Jan-q1-75	fulldisclosure-2016-May-33.txt	108	Wolfgang feedyourhead at	68	0.371	5	3
Fulldisclosure-2015-Feb-q1-53	nmap-dev-2017-q2-8.txt	54	Scott Arciszewski	1,304	0.370	3	2
Fulldisclosure-2015-Feb-q1-90	fulldisclosure-2016-Aug-118.txt	83	Praveen D	1,314	0.451	1	1
Bugtraq-2017-Jan-q1-18	bugtraq-2016-Jan-146.txt	125	Vulnerability Lab	3,419	0.677	13	3
Bugtraq-2017-Jun-q2-56	fulldisclosure-2015-May-52.txt	130	SEC Consult Vulnerability Lab	3,829	0.532	13	3
Bugtraq-2016-Jan-q1-75	nmap-dev-2015-q2-40.txt	79	Slackware Security Team	4,009	0.332	2	1
Bugtraq-2017-Apr-q2-158	nmap-dev-2017-q2-8.txt	54	Salvatore Bonaccorso	4,215	0.432	1	1
Nmap-dev-2017-Mar-q1-226	bugtraq-2016-Apr-36.txt	65	Henri Doreau	4,831	0.533	3	2
Nmap-dev-2015-Nov-q4-107	fulldisclosure-2016-May-33.txt	108	Peter Houppermans	6,849	0.600	8	3
Nmap-dev-2015-Oct-q4-63	nmap-dev-2015-q4-276.txt	68	Mark Scrano	6,853	0.496	1	1
Pen-test-2017-Jul-q3-1	bugtraq-2017-Jul-8.txt	63	Hafez Kamal	7,994	0.252	3	2
Pen-test-2016-Feb-q1-2	nmap-dev-2017-q2-8.txt	54	Francisco Amato	8,071	0.357	2	1
Pen-test-2016-Dec-q4-0	bugtraq-2017-Mar-39.txt	115	ERPScan inc	8,072	0.521	9	3

In the test scenario, we investigated incident reports from 'seclists' CS feed to classify those by threat level and to automatically manage them from acquisition to disposal without involvement of human analyst (see Table II). Due to the large number of results in this table, we describe only selected classification results, which

demonstrate typical cases. Query incident ID in 'seclists' terms is presented in the first column. The second column shows the first of the detected related incident IDs. The similarity score for found related incidents for selected examples is nearly 1.0. In the third column, we show a number of detected common words between query and found incidents. Column 'Source' depicts an incident source that can be a person or an organisation. The next four columns are related to Smart Contracts and show assigned Blockchain ID, consumed time, number of significant terms and threat level. The experimental results are represented in Figure 4 and show the distribution of threat incident reports over the last three years, respective of high, middle, and low threat levels. Each incident category is flagged by an assigned colour. The Y axis is a range of the number of incidents and the X axis is a time scale split into quarters. The figure shows that the most productive category for high (up to 325) and low (up to 215) threats is a 'bugtraq' category, whereas 'nmap-dev' (93) and 'fulldisclosure' (97) are dominating middle threat reports. For a given period of time, most active phase for all levels is from 'Q4-2015' to 'Q3-2016'. Visualization of incident reports provides an analyst with a quick and descriptive SA picture. To focus on a particular area, the analyst can perform fine tuning, adjust the time scale or select a particular category or source.

FIGURE 4. PLOT FOR DISTRIBUTION OF THREAT INCIDENT REPORTS OVER LAST THREE YEARS FOR DIFFERENT THREAT LEVELS SHARED QUARTERLY.



As a use case scenario, assume that SOC has received an incident report from Vulnerability Lab in January 2017. On receiving this report, our Smart Contract triggers automatic analysis and classification of this incident report. According to Table 2, we see that this incident is assigned a Smart Contract identifier 3419 and the contract identifies 13 significant terms. Going through the contract logic we estimate both the regular and the weighted threat level as a 'high threat' (3). That means it should be handled soonest and with highest priority. The incident is automatically tagged and enriched with additional data from CS feeds and tools. Links to similar

incidents are established. All this facilitates the triage process for a cyber analyst and performs analysis steps that are usually done manually. According to the evaluated classification level, Smart Contract defines timestamps for automated archival and disposal of incident data. Therefore, a cyber analyst does not need to worry about the incident life-cycle and can focus their resources on triage for urgent cases.

The smallest duration for one Smart Contract operation was 0.252 seconds from Blockchain ID 7994 report and the longest operation time 0.677 report with ID 3419. This difference can be explained by the different report sizes (we calculate hash for report content) and different risk points numbers (3 for ID 7994 vs. 13 for ID 3419). This evaluation also gives a simple overview of detected significant terms, such as 'attack', 'hack', 'phishing' for high threat incidents, 'access', 'authentication', and 'encode' for middle threat incidents and 'key', 'capability', and 'investigation' for low level threats. Having a Smart Contract ID, the analyst is able to retrieve status data of a particular incident report from Blockchain using Smart Contract (e.g. by hash, provenance, time, tags, owner).

TABLE 3. OVERVIEW ABOUT AGGREGATED THREAT REPORTS FOR DIFFERENT THREAT CATEGORIES.

Threat Category	High Threat	Middle Threat	Low Threat	Total
Fulldisclosure	724	558	590	1,872
Bugtraq	758	147	542	1,447
Pen-test	55	43	4	102
Nmap-dev	430	674	1,325	2,429
Sum	1,967	1,422	2,461	5,850

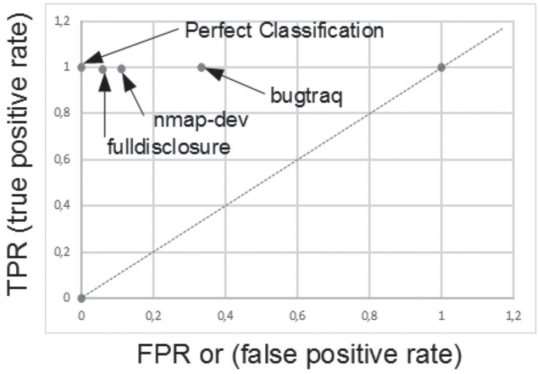
The category overview experimental results are presented in Table 3 which shows the distribution of high, middle and low threat level incidents for different incident categories. This table demonstrates that most incident reports (2,429) come from the 'nmap-dev' category, followed by 'fulldisclosure' (1,872) and 'bugtraq' (1,447). Most of incident reports belong to the low threat level (2,461) but the report number classified as high threat is also high (1,967). Most high threat level reports come from the 'fulldisclosure' (724) and 'bugtraq' (758) categories. That means that these categories should be addressed first by incident management.

C. Evaluation Effectiveness

We can see that, in general, the autoencoder training accuracy improves with every iteration (epoch) from 0.674 at the beginning to 0.942 at the end, which is sufficiently good; whereas training loss (error) of original information decreases from 0.691 to 0.152. This means that the decompressed outputs will be degraded compared to the

original inputs, but it is an acceptable rate. Similarly, validation accuracy is in the range between 0.616 and 0.915. Validation loss decreases from 0.684 to 0.220.

FIGURE 5. ROC SPACE PLOT.



The classification effectiveness for high priority incidents can be determined in terms of a Relative Operating Characteristic (ROC) using the labelled ground truth query dataset. SA analysis divided the provided incident reports into two groups: 'high' and 'low priority' by associated expert parameters and thresholds for each category; e.g. for the 'fulldisclosure' category the provided algorithm detected 229 true positive incidents, 14 true negative reports, one false positive incident and two false negative documents. The primary statistical performance metrics for ROC evaluation are sensitivity (0.991) or true positive rate and false positive rate (0.059). The associated ROC value is represented by the point (0.059, 0.991). The ROC space (see Figure 5) demonstrates that the calculated FPR and TPR values for the evaluated categories are located very close to the so called perfect classification point (0, 1). The calculation results demonstrate that the calculated similarity score values for the query documents are located very close to the labelled classification. These results demonstrate that an automatic approach for cyber incident classification of the method described is very effective and is a significant improvement on manual analysis. Therefore, an analysis method based on deep autoencoder techniques can be suggested as an effective method for incident classification, and as a supporting method to establish cyber SA. The results of the analysis confirm our hypothesis that an automated approach is able to reliably classify incidents, thus making analysis of a large number of cyber incidents a feasible and affordable process. However, further research is required to improve the decision and accuracy metrics of this method.

6. CONCLUSIONS

In this work, we have presented an automated approach to classify and manage incident reports for establishing cyber situational awareness using a deep autoencoder neural network for classification and a Smart Contracts technique provided by Blockchain technology for incident management. The developed system should assist cyber analysts by protecting Critical Infrastructures against increasing cyber threats. The main contribution of this work is a real-time solution that could replace human input for a large number of cyber incident analysis tasks in order to facilitate cyber incident classification, eliminate irrelevant information and focus on important information to promptly perform mitigation steps. Another contribution is the use of the Smart Contract techniques to provide an automated trusted system for an incident management life-cycle that allows automatic acquisition, classification, use, archiving, and disposal. An additional advantage of this approach is a reduction of human analysis costs. Ultimately, our research will lead to the creation of automated security assessment tools with more effective handling of cyber incidents.

REFERENCES

[1] P. Barford et al., 'Cyber SA: Situational Awareness for Cyber Defense,' in *Cyber Situational Awareness, Advances in Information Security*, vol 46, Springer, Boston, MA, 2010.

[2] A. Kott and C. Wang, *Cyber Defense and Situational Awareness*, Switzerland: Springer Int. Publ., volume 62, ISBN 978-3-319-11391-3, 2014.

[3] G. Wood, 'Ethereum: A Secure Decentralised Generalised Transaction Ledger,' in *EIP-150 REVISION*, http://gavwood.com/paper.pdf, 2014.

[4] G. Settanni, F. Skopik, R. Graf, M. Wurzenberger, and R. Fiedler, 'Correlating cyber incident information to establish situational awareness in Critical Infrastructures,' in *14th Annual Conference on Privacy, Security and Trust (PST)*, Auchland, New Zealand, pp. 78-81, 2016.

[5] L. Samarji, 'Coordination and Concurrency Aware Likelihood Assessment of Simultaneous Attacks,' in *Third International Conference on Security and Privacy in Communication Networks SecureComm*, vol. 152, pp 524-529, 2015.

[6] T. Reguly, 'Does Anybody Really Care About Vulnerability Scoring?,' in *International Conference on Computational Science and Engineering*, 2013.

[7] L. Maghrabi, E. Pfluegel, L. Al-Fagih, R. Graf, G. Settanni, and F. Skopik, 'Improved software vulnerability patching techniques using CVSS and game theory,' in *International Conference on Cyber Security And Protection Of Digital Services (Cyber Security)*, pp. 494-505, London, 2017.

[8] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, 'Semi-supervised Recursive Autoencoders for Predicting Sentiment Distributions,' in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, pp. 151-161, Edinburgh, Scotland, UK, 2011.

[9] A. Krizhevsky and G. E. Hinton, 'Using very deep autoencoders for content-based image retrieval.,' in Proceedings *ESANN*, Bruges, Belgium, 2011.

[10] Y. Bo, X. Zong-ben, and L. Cheng-hua, 'Latent semantic analysis for text categorization using neural network,' in *Knowledge-Based Systems*, volume 21, number 8, pp. 900-904, 2008.

[11] S. L. Y. Lam and D. L. Lee, 'Feature reduction for neural network based text categorization,' in *Proceedings. 6th International Conference on Advanced Systems for Advanced Applications*, pp. 195-202, Hsinchu, 1999.

[12] L. Auria, 'Support Vector Machines (SVM) as a Technique for Solvency Analysis,' in *DIW Berlin*, Paper 811, 2008.

[13] S.-S. Weng, 'Ontology construction for information classification,' in *Exp. Systems with Applications*, volume 31, number 1, pp. 1-12, 2006.
[14] C. H. Caldas and L. Soibelman, 'Automating hierarchical document classification for construction management information systems,' in *Automation in Construction*, volume 12, number 4, pp. 395-406, 2003.
[15] M. Uma and G. Padmavath, 'A Survey on Various Cyber Attacks and Their Classification,' in *International Journal of Network Security*, Coimbatore, volume 15, pp. 390-396, 2013.
[16] S. Harris and F. Maymi, *CISSP All-in-One Exam Guide*, book, New York: McGraw-Hill Education, 2016.
[17] G. W. Peters, *Understanding Modern Banking Ledgers Through Blockchain Technologies: Future of Transaction Processing and Smart Contracts on the Internet of Money*, Springer Int. Publishing, pp. 239-278, 2016.
[18] M. Macdonald, L. Liu-Thorrold, and R. Julien, 'The Blockchain: A Comparison of Platforms and Their Uses Beyond Bitcoin,' in *COMS4507 - Adv. Computer and Network Security*, Univ. of Queensland, 2017.