HOSTED BY
ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/jides

Using neural networks to aid CVSS risk aggregation — An empirically validated approach

Alexander Beck[a], *Stefan Rass*[b,*]

[a] *VW Financial Services AG, Gifhorner Strasse 57, Braunschweig, Germany*
[b] *Universität Klagenfurt, Universitätsstrasse 65-67, 9020 Klagenfurt, Austria*

HIGHLIGHTS

- A method for automated CVSS risk aggregation is proposed.
- The aggregation can be tailored/trained to domain expertise and uncertain knowledge.
- Results have been verified along an empirical study.
- A method to reduce answer variability and ambiguity in empirical CVSS risk assessments is described.

ARTICLE INFO

Article history:
Received 14 July 2016
Accepted 31 October 2016
Published online 23 November 2016

Keywords:
Risk management
Neural network
Data aggregation

ABSTRACT

Managing risks in large information infrastructures is often tied to inevitable simplification of the system, to make a risk analysis feasible. One common way of "compacting" matters for efficient decision making is to aggregate vulnerabilities and risks identified for distinct components into an overall risk measure related to an entire subsystem and the system as a whole. Traditionally, this aggregation is done pessimistically by taking the overall risk as the maximum of all individual risks, following the heuristic understanding that the "security chain" is only as strong as its weakest link. As that method is quite wasteful of information, this work proposes a new approach, which uses neural networks to resemble human expert's decision making in the same regard. To validate the concept, we conducted an empirical study on human expert's risk assessments, and trained several candidate networks on the empirical data to identify the best approximation to the opinions in our expert group.

1. Introduction

Risk management is among the core duties of the general steering in large companies. While financial risk management enjoys a comprehensive set of helpful tools and methods, security risk management until today appears to widely rely on heuristics, (subjective) human expertise and common practice knowledge. Likewise, compiling vulnerabilities, known problems and security issues of components into a concise risk report for decision making

Peer review under responsibility of Qassim University.
* *Corresponding author.*
E-mail addresses: alexander.beck@vwfs.com (A. Beck), stefan.rass@aau.at (S. Rass).

http://dx.doi.org/10.1016/j.jides.2016.10.002

is a matter of simplifying and quantifying a situation and its impact, to make security manageable. Especially the quantification is herein a central and crucial issue, as security is a cost-benefit consideration, and quantitative measures of security are hard to define soundly. Most of the related difficulty comes from the inherent complexity of contemporary information and communication technology (ICT), which makes a hierarchical decomposition of a system into smaller subsystems necessary. Upon such a decomposition, a risk assessment can be applied, either top-down (in which case the overall risk is broken up into individual risks for subsystems), or bottom-up, when risks that are specific for limited scopes are put together into a risk picture of the bigger system. This aggregation is then iterated along the hierarchical decomposition up to the top, where the final result on the risk can be reported to decision makers for the daily business of risk control. Unfortunately, the precise process of how to aggregate risks is rarely well documented nor comprehensively studied or understood (from a psychological perspective), so most of this labor is done using rules-of-thumb. More importantly, the specific ways in which risk is aggregated is often quite context dependent. Today, these dependencies have led to a large volume of best-practices relating to many diverse domains. Risk management standards are in their core a compilation of such best practices that have been abstracted to make it amendable to the specific situation at hand. This work is an extended version of Beck and Rass [1], where a first step towards a general and flexible risk aggregation rule has been proposed. One of the few related existing such general rules to aggregate risks is the "maximum principle" (cf. section 4.3.3. in BSI [2]), which prescribes to take the vulnerability of a (sub) system as the maximum vulnerability of any of its components (herein, "vulnerabilities" are quantified as likelihoods for failure upon any attack from a known and a-priori identified set of threats).

Obviously, this approach is wasteful on information and pessimistically overestimates the risk, so that risk experts tend to refine a so-obtained first guess using their own expertise and experience. The problem that motivated this research, was an automated aid for risk assessment and decision support by "approximating" human decision making. We propose doing so by using neural networks (alternatives are discussed in Section 1.2). Our contribution is a concrete neural network (NN) trained on empirical findings from a study that queried risk experts on several scenarios, asking for their informed opinion about the overall risk as they would assess it in a real process.

1.1. *Motivation by example*

As a simplified example, consider a subsystem in an enterprise infrastructure model, composed from two representations, given as Figs. 1 and 2. First, we have a physical dependency model of applications on components (Fig. 1), which is augmented by the logical dependency model of applications on one another (Fig. 2). The risk analysis is usually done in a bottom-up fashion. That is, the vulnerability of application A is influenced by the security of its (indirect) ancestor nodes VM_1, VM_2 and their parent AS_2. Normally, we need

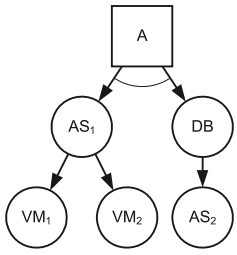

Fig. 1 – Dependencies of applications on physical components.

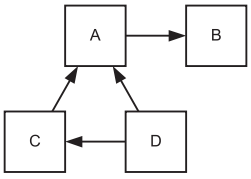

Fig. 2 – (Logical) Interdependencies between applications.

to account for "and/or"-dependency relations, if an application depends on any ("or") or all ("and") shown components. Various industrial standards can help with the assessment, and our pick in this work is the common vulnerability scoring system (CVSS; see first.org[3]). Let CVSS(X) denote the 12th dimensional (real-valued) scoring assigned to component X that results from the expert rating the CVSS criteria related to component X in terms of CVSS.[1] So, the risk assessment on application A would start with $\text{CVSS}(VM_1)$, $\text{CVSS}(VM_2)$. These two vectors would then go into the assessment $\text{CVSS}(AS_1)$. However, the assessment *cannot* straightforwardly take the maximum of the children's assessments (in a naive attempt to model the "OR-branch" of AS_1 into VM_1, VM_2), since the expert has to take into account switching times between the working and the fallback virtual machine, as well as characteristics of AS_1 that are intrinsic to the application server itself. Therefore, the assessment $\text{CVSS}(AS_1)$ only partially but not exclusively depends on $\text{CVSS}(VM_1)$ and $\text{CVSS}(VM_2)$. At this stage, most standard risk management methods hit their limits and leave the consideration of the relevant information up to the expert. In our case, this means casting the scores $\text{CVSS}(VM_1)$, $\text{CVSS}(VM_2)$ and the information known about AS_1 into a scoring $\text{CVSS}(AS_1)$. Normally, this is a nontrivial and fuzzy process.

Abstractly, the risk expert's task is traversing the graph bottom-up, where at node AS_1, his duty is to evaluate $\text{CVSS}(AS_1) = f(\text{CVSS}(VM_1), \text{CVSS}(VM_2))$, additional information about AS_1, where the function f here represents her/his expertise, experience and general/personal method to assess the vulnerability for the application server AS_1. This process is nontrivial to automate, since it assumes the graph to be acyclic, and a straightforward bottom-up aggregation would implicitly assume each node to appear exactly once in the

[1] Note that CVSS does only address confidentiality, integrity and availability. Accounting for Authenticity and other security goals is up to a manual addition to the risk management process that we do not discuss here.

graph. Since a particular component may be vital to many other components (and vice versa), we have to avoid double-aggregation of its risk into the overall picture.

For the sake of comparison and consistency (also between scorings of different systems, say, if the decision concerns the selection of one out of several candidate system offers), we can reasonably assume that the risk expert is obliged to make her/his assessment using a (subjective but) *fixed* method f, whose outcome does only depend on the information available on the system. The method itself may, however, not change between different assessments, as this would defeat the purpose of CVSS being also a *comparative* scoring system.

Our contribution in this work is exactly how to transfer an expert's risk assessment and aggregation method f into an NN, for a threefold benefit: first, we equip the expert with automated tool support that is tailored to her/his knowledge, expertise and experience. Second, we assure consistency among and thus comparability between all assessments (as there may be very many in complex infrastructures). Third, we make the expert's risk aggregation service available to others, thus allowing to delegate these decisions upon the so-achieved tool-support.

1.2. *Related work*

It is quite noticeable that methods of artificial intelligence have not yet seen much application for decision support in the security domain, besides only a few exceptions: Kai Sun et al. [4] for example, show how risk assessment of a power-supply utility network can be done, based on attributes assigned to the components of the system. Practically, such assessments are quite similar to those in IT infrastructures, with the major difference being the geographic span of the system. In this reference, the authors use decision-trees on presumed discrete attributes to derive an assessment of the overall system (in a hierarchical fashion, similar as we propose here). In reality, however, security assessments do not exclusively depend on discretizable attributes, and to a significant extent rely on expertise and experience of the assessor. Thus, a decision-tree approach would encounter difficulties due to vague inputs being required, and due to the necessity of accounting for interdependencies among components (which would go into the assessment via the aforementioned expertise and experience). An NN is thus appealing for its ability to learn from data, which spares the human expert a "formalization" of one's own methods.

Fuzzy logic has been designed to let humans speak out their heuristics in natural language terms, while directly producing formulae to do reasoning on that ground. While appealing in many contexts, risk aggregation is a process that is often perceived as being difficult to define rules for. More importantly for our specific application of CVSS is the absence of linguistic variables, since the scoring is a crisp number that would need to be fuzzified first (perhaps somewhat unnecessarily).

These challenges were independently discussed by McCalley et al. [5], who in their paper seek a deterministic security assessment method, but back then already identify the need for tool- and decision-support to tackle this complex task. Moreover, this reference is among the first to recognize that a single "measure of security" is insufficient, which justifies the use of higher-dimensional metrics like CVSS and neural networks to do classifications and aggregations in a highly nonlinear manner. Both call for an account of the "whole picture" (rather than focused local analyses). We naturally serve this need, as the output of the CVSS aggregation can easily be cast into color-indicators of severity, thus offering a graphical visualization of where problems in a system are most likely located. Although this related reference also proposes this, their approach lacks an automated assessment and still leaves the final aggregation task up to a human expert; a gap that our contribution may close. Relevant standards such as NIST [6] explicitly prescribe risk aggregation, but also leave the details mostly unspecified, thus calling for development of aggregation methods. The need to do so has a long history, substantiated for example by Blakley [7], Carroll [8], but also in different fields of risk management, say the financial sector, where NNs and support vector machines are used to analyze financial risks (see Bol et al. [9] or Yu et al. [10]). The field of security metrics and how to work with them is very active, with a vast number of different approaches having been defined; see Savola [11], Ming et al. [12], or Hayden [13] and references therein, to mention only a few. Some of these are specially tailored solutions (such as Ming et al. [12]) or general overviews with huge collection of heuristics and best practices; such as HEISC [14] or Payne [15]. Common to all these recommended methods is their usual lack of tool-support and leaving much of the labour up to human experts. This work is a step towards automating the aggregation process, which is among the stages where most human expertise is required.

Another related approach goes for an analytical model, which divides the infrastructure into (three) perspectives of physical components, the user and the services. This division is more general than ours, but includes the user's perception of risk in the assessment, which is not relevant for an internal assessment normally (and thus excluded from our considerations here). Finally, Bayesian reasoning appears as a natural candidate for an analytical approach to aggregating risk. The issue here is, however, that convergence of Bayesian estimators hinges on massive amounts of data to be trained on. Ironically, in the security context, getting more data requires more secure incidents, which is exactly what risk management shall prevent.

2. The empirical study

We asked a set of 50 experts for their (subjective) risk assessments, based on three different scenarios based on real-life experience. Practical experience shows that subjective difference in expert opinions may be significant, despite the scope of the assessment being precisely defined and narrowed. Thus, asking experts for a pure CVSS assessment digs up answers that may be mutually inconsistent and have a strong variability in the answers. To reduce this to the end of a "more robust" assessment result, we defined 10 additional questions on top of the subsequent CVSS scoring—we refer to this set of questions as the "meta-metric". Besides the purpose of reducing the

answer variability, this also provides additional insights on how experts reach their votes. The results were anonymized and the study returned a total of 45 records, from which 75% were randomly chosen for the training, and the remaining 25% were used for verifying the network performance.

Before reporting the results in Section 3, we first describe the particular questions that the meta-metric consists of: Each of the following questions was to be answered in the familiar terms as the subsequent CVSS, including the option "not defined" (ND), if the question does not apply:

1. Protective Target (confidentiality, integrity, availability/ND): which is the security goal of interest? During interviews, it turned out that the subjective vulnerability assessment is strongly influenced by whether the expert rates the scenario in terms of confidentiality, integrity or authenticity. Part of the variation in the data may thus be due to the different presumptions among experts, which the first question in the meta-metric shall equalize.
2. Redundancy (yes/no/ND): is the system of interest unique or is there a backup-copy of it? If so, then the redundant system may not have the same patch level, thus vulnerabilities that are not found in one system may indeed be found in its fallback version. Thus, the risk assessment should be on the "weakest" version among all redundant instances.
3. Application type (input–output/input–output-processing/input–output-processing-storage/ND): Again, interviews indicated that experts also differentiate systems according to whether data is only transmitted, stored, processed, or if combinations thereof occur. Remaining unspecific about such details leaves individual experts to their own assumptions, whose equalization by this a-priori question helps reducing the variations in the answers.
4. Data type (payload/meta-information/ND): often, risk is quite different depending on whether the information itself or only meta-information is relevant. For example, encryption prevents access to the payload, but does not hide who is talking to whom. The latter is meta-information that can as well be relevant for risk and vulnerability assessment (likewise, an integrity protection may protect against manipulation of transmitted data, but it can nevertheless be unwantedly re-routed by altering the unprotected meta-information).
5. Usage (single/multiple/ND): depending on whether a system or component is relevant for only one or perhaps many applications greatly affects its risk assessment. Therefore, it is useful to have users think about this detail a priori.
6. Security node (yes/no/ND): some components are designated to the business workflows, while others are for pure security services. For example, a firewall has a purely security-relevant role, as opposed to a file-server, which is required for the daily business. The vulnerability assessment of the two is, however, quite different, since the outage of a firewall may be compensated by a redundant network connection, whereas the file server being down may cause much more severe business interruption.

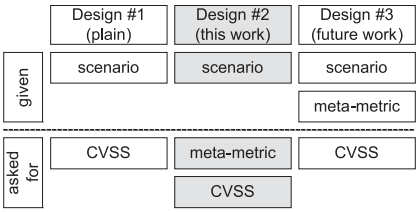

Fig. 3 – Possible designs for the empirical study.

7. Proprietary (yes/no/ND): do the sub-systems provide standard or "hand-crafted" individual solutions? Self-made mechanisms are expected to be much less well-tested and may have vulnerabilities that proprietary standard systems may not have (the phenomenon is similar to Kerckhoffs' principle, which prescribes cryptographic systems to be published in their full details. Consequently, these "open" systems enjoy the interest of the entire scientific community, so that vulnerabilities and weaknesses are identified much earlier than otherwise).
8. Open source (yes/no/ND): open source systems are usually even better tested than proprietary systems and have a short patch interval on average. However, the primary interest of the community is on functional correctness, rather than security functionality. Thus, although open source usually provides high-quality systems, a security auditing or code review for security is quite difficult there, since the design documentation is usually unavailable. Hence, and somewhat ironically, although the systems are open-source, they often appear as effective "black boxes", simply due to their high complexity, which makes them almost inaccessible to a deep-level analysis.

These questions have been added to the CVSS scoring, so that we obtained a total of $10 + 12 = 22$ answers. An alternative design of the study could give the meta-metric to the expert a priori, thus turning it into auxiliary information rather than an auxiliary questionnaire. Such a setup could be used to measure the impact of the meta-metric, relative to a plain CVSS assessment. Hence, among the three basic designs that were possible for the study, shown in Fig. 3, we explicitly focus on the second of the designs in this work, leaving the third one as a route for future work on the quality of the meta-metric itself (design #3 of the study).

3. Training the neural networks

We chose a perceptron configuration, trying to train networks with one or two hidden layers. The number of nodes in the hidden layer has been determined from various heuristic rules. Using CVSS, a risk assessment consists of twelve scores (assigned by the expert) and ten additional questions that were introduced for this work only to refine the results. Towards aggregating two such extended CVSS assessments into a single (plain) CVSS scoring in twelve dimensions, our NN has $2 \times (12 + 10) = 44$ input nodes, and 12 output nodes.

We trained (using resilient propagation learning; cf. Anastasiadis et al. [16]) and tested a total of 13 networks, whose structure and performance results are reported in

Table 1 – Neural network training results.

No of hidden layer	Size of hidden layer	bias?	E	N	validation
1	12	no	0.01422627	-	×
1	12	yes	0.013914681	-	×
1	16	no	0.000999515	51,103	✓
1	16	yes	0.000997663	14,779	✓
1	20	no	0.000999628	13,892	✓
1	20	yes	0.000999795	5,955	✓
1	28	no	0.000999054	4,392	✓
1	28	yes	0.000999135	3,544	✓
1	36	no	0.000999939	4,340	✓
1	36	yes	0.000999672	3,806	✓
2	12 + 12	no	0.00859552	-	×
2	12 + 12	yes	0.005936428	-	×
2	16 + 16	no	0.003361971	-	×

Table 1. The best network in our experiments was feed forward and had 16 hidden nodes in a single hidden layer, together with a bias neuron connected to all nodes in the hidden layer, and using a hyperbolic tangent as activation function for all nodes (the respective row in Table 1 is highlighted). Weights were assigned to all inner edges, except for the output edges.

Table 1 is to be read as follows: besides the description of the concrete topology, we evaluated the error rate E after 10.000 iterations, as the ration between network output and the expected result of the test set, counting the number N of iterations until the learning algorithm converged towards an error rate below 0.001. The training was done on a subset of 75% of the expert test-cases, with the remaining 25% being validation data, on which the result from the network was checked against (i.e., compared to the approved expert's opinions). We noted "successful" if the value E ("error rate E after 10.000 iterations") of successful such verification among all trials was below 0.001. In this case we have an automatically approximated result of the test-set, so that we can see the results of the NN with high accuracy corresponding to a manual review. An interesting question of future work may concern the use of other network topologies not covered in Table 1, say feedback networks or ones with more than two hidden layer. Overfitting is, however, an issue to be avoided here, so we leave this direction unexplored in this work.

Using the network for hierarchical aggregation is then a matter of feeding a CVSS risk assessment with meta-metric into the NN, and then re-using its output as input to the NN in the next stage (we revisit this in more detail in the next section). Note that the meta-metric in this "inner" aggregation stage may indeed be much more efficient and easy to specify than the regular CVSS metric on the physical entities under consideration. For example, the protective target (say, "confidentiality") will remain the same over the entire risk aggregation; likewise, the meta-metric's attribute "redundancy" may as well be an identical input to all aggregations along the way for an entire redundant subsystem. For a hierarchical aggregation in a given system, the network can be used to do the aggregation in each step. However, it must be stressed that the network should in any case be repeatedly adapted and re-trained in order to remain a reasonable approximation to the overall aggregated risk. Indeed, security audits are in any case to be repeated, and security certificates naturally expire (for ISO 27000, the maximal validity period is three years). It appears thus advisable to use the independent risk assessment that may be provided along a security audit or certificate renewal (for validation of the existing information risk management system in the enterprise), to re-train the network towards updating its risk assessment. As such, the network training should become part of the ongoing security life-cycle, and in case of ISO 27000, the PDCA cycle (plan-do-check-act).

4. Integrating the network in the decision process

With the automated aggregation in place, we can now partially automate a decision process with help of NN-based risk aggregation along hierarchical aggregation. The NN plays the role of the function f in the bottom-up traversal.

A crucial point here is the automated account for interdependencies, which is also a central requirement in risk management decision making. This interdependency comes into the NN through the expert training data, and therefore does not have to be modeled explicitly (as would perhaps be necessary for other approaches). Now, integrating the NN as a substitute tool at the point where the expert would be required to aggregate risks manually, we end up with a widely automatic procedure to reach a risk assessment for the overall system, which can be presented to a decision maker.

Summarizing this procedure, let us assume that there is a hierarchical decomposition of the infrastructure into applications that (recursively) depend on others, until the bottom of the hierarchy, where the physical system components are located (cf. Figs. 1 and 2 as examples).

The risk aggregation process then proceeds upwards by invoking the neuronal network for the aforementioned aggregation function f (cf. the motivating example in the introduction) so as to layer-by-layer aggregate risks up to the top. It is exactly the f-operation where the human expert would be required otherwise. Fig. 4 displays the point where the NN is integrated in the risk management process, showing how the process helps to "tool-aided" the risk assessment.

In practice, the aggregation network should (must) be adapted to the particular context of an application, since risk aggregation may look different depending on the system at hand. Moreover, NNs do not answer the "why" of a particular aggregation result, which, however, may rarely be necessary since the NN is trained to approximate human reasoning to the best possible extent. This drawback of the NN is general, but nonetheless appears to hardly limit the applicability and usefulness of the aggregation in the daily business of security risk assessment. The primary advantage of having the risk assessment partly automated is the possibility of "playing around" with different configurations towards getting a feeling on which investments may be better than others. So, the security officer as the primary user of this

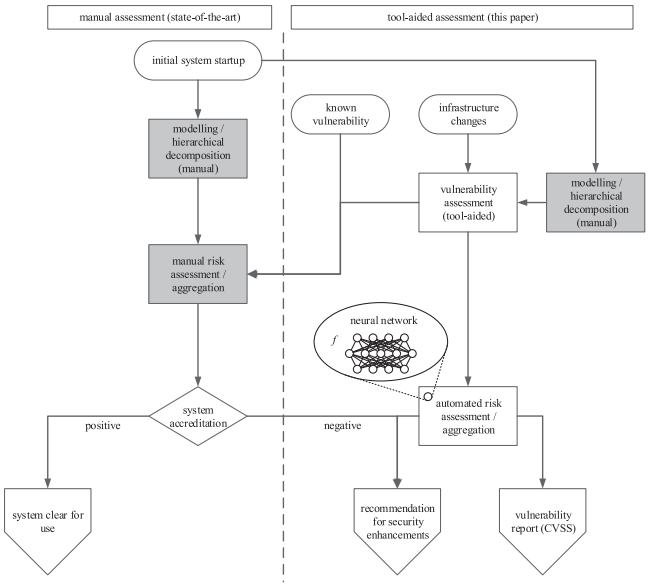

Fig. 4 – Manual vs. automated risk assessment process.

risk aggregation may use the NN on a variety of different hypothetical (possible) adaptions to the system, to the end of figuring out the best choice among a set of improvements that fall into the available budget. The question of the "why" then only applies to the best solution found in this way, by which the security officer can resort to her/his own expertise to explain the effects of the envisioned improvements. Its results are obtained by the NN, on grounds of prior training based on expert opinions. When confronting a decision maker, arguments may thus be easier to settle since there is a concrete proposal being explained by a domain expert, with implications and effects being underpinned by the compound knowledge of several experts (whose input is embodied in the NN).

5. Conclusions and outlook

Although the task of risk assessment in general and risk aggregation in particular is usually widely based on human expertise, surprisingly little effort has so far been put on mimicking human reasoning within the standardized risk assessment processes. Tool support is particularly rare in this regard, and artificial intelligence techniques seem to offer an invaluable contribution to the recognized need for decision support for risk managers. This work analyzed NNs for the purpose of risk aggregation, by proposing to capture an expert's intuitive aggregation heuristic into a neural network. The network is thus designed to resemble human decision processes as close as possible, to the end of taking the duty of risk aggregation from the human expert. This automation and flexibility comes at the price of different networks arising from different expert trainings. To retain comparability of scorings, we thus recommend to train networks on a compound data set collected from multiple domain experts, so that the NN is not fitted to a single person, but a group of people. Among several possible candidate network topologies that were trained, we identified a perceptron with one hidden layer to perform best on the CVSS risk aggregation problem.

The overall benefit of the proposed method lies in the ease of integration into standardized processes for partially automated security risk management support (see Fig. 4 for a comparison). As we were using the common vulnerability scoring system as our running example here, future work may as well target other such rating schemes for risk aggregation, to extend the capabilities of these (and other) techniques from artificial intelligence to the IT security area. Very little has been done in this direction so far, but the indications found in this work point this out as a promising direction for the future.

Acknowledgment

We are indebted to A. Heidorn for implementing the network training and delivering the data reported in Table 1 [17]. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

REFERENCES

[1] A. Beck, S. Rass, Security risk aggregation based on neural networks – An empirically validated approach, in: Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference, (FLAIRS), AAAI, 2016, pp. 294–297.

[2] Federal Office for Information Security (BSI), Germany; BSI Standard 100-2 IT-Grundschutz Methodology, Version 2, 2008. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/BSIStandards/standard_100-2_e_pdf.pdf?__blob=publicationFile.

[3] Forum of Incident Response and Security Teams (first.org); Common Vulnerability Scoring System CVSS v3.0 Specification, 2015. https://www.first.org/cvss/cvss-v30-specification-v1.7.pdf.
[4] Kai Sun, S. Likhate, V. Vittal, V.S. Kolluri, S. Mandal, An online dynamic security assessment scheme using phasor measurements and decision trees, IEEE Trans. Power Syst. 22 (4) (2007) 1935–1943.
[5] J.D. McCalley, V. Vittal, N. Abi-Samra, An overview of risk based security assessment, in: Power Engineering Society Summer Meeting, Vol. 1, IEEE, 1999, pp. 173–178. no., 18–22 Jul.
[6] National Institute of Standards and Technology (NIST), Special Publication 800-30, Revision 1, Guide for Conducting Risk As-sessments, September 2012.
[7] Bob Blakley, Ellen McDermott, G. Dan, Information security is information risk management, in: Proceedings of the 2001 Workshop on New Security Paradigms, (NSPW'01), ACM, New York, NY, USA, 2001, pp. 97–104.
[8] G. Carroll, Enterprise Compliance Today – How to aggregate risk in an Enterprise Risk Management (ERM) system, November 2013, [online] http://www.fasttrack365.com/blog/bid/347034/How-to-aggregate-risk-in-an-Enterprise-Risk-Management-ERM-system (accessed on 11.11.15).
[9] G. Bol, G. Nakhaeizadeh, and K.-H. Vollmer (Eds.), Risk measurement, in: Econometrics and Neural Networks – Selected Articles of the 6th Econometric-Workshop in Karlsruhe, Germany, Physica, Contributions to Economics, 1998.
[10] L. Yu, S. Wang, K.K. Lai, L. Zhou, Bio-Inspired Credit Risk Analysis – Computational Intelligence with Support Vector Ma-chines, Springer, 2008.
[11] R.M. Savola, Towards a taxonomy for information security metrics, in: Proceedings of the 2007 ACM Workshop on Quality of Protection, (QoP'07), ACM, NY, USA, 2007, pp. 28–30.
[12] Ni Ming, J.D. McCalley, V. Vittal, T. Tayyib, Online risk-based security assessment, IEEE Trans. Power Syst. 18 (1) (2003) 258–265.
[13] Lance Hayden, IT Security Metrics: A Practical Framework for Measuring Security and Protecting Data, Mcgraw-Hill, 2010.
[14] Higher Education Information Security Council (HEISC), Infor-mation Security Guide: Effective Practices and Solutions for Higher Education, 2014, [online] https://spaces.internet2.edu/display/2014infosecurityguide (accessed on 11.11.15).
[15] S.C. Payne, A Guide to Security Metrics, 2006, [online] http://www.educause.edu/library/resources/guide-security-metrics (accessed on 11.11.15).
[16] A.D. Anastasiadis, G.D. Magoulas, M.N. Vrahatis, New globally convergent training scheme based on the resilient propagation algorithm, Neurocomputing 64 (2005) 253–270.
[17] A. Heidorn, Prototypische Implementierung eines Security Risk Assessment Frameworks (SRAF) zur Erstellung und Aggregation von SRAF Graphen (Bachelor Thesis), Wernigerode, Hochschule Harz - BA, 2014.