Proceedings of the Twenty-Ninth International
Florida Artificial Intelligence Research Society Conference

Security Risk Aggregation Based on Neural Networks – An Empirically Validated Approach

Alexander Beck[1], Stefan Rass[2]

[1] VW Financial Services AG, Gifhorner Strasse 57, Braunschweig, Germany
[2] Universität Klagenfurt, Universitätsstrase 65-67, 9020 Klagenfurt, Austria
alexander.beck@vwfs.com, stefan.rass@aau.at

Abstract

Managing risks in large information infrastructures is a task that is often infeasible without proper simplification of the system. One common way of "compacting" matters towards easing decision making is to aggregate vulnerabilities and risks identified for distinct components into an overall risk measure related to an entire subsystem. Traditionally, this aggregation is done pessimistically by taking the overall risk as the maximum of all individual risks ("the 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.

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 rely mostly on heuristics and (subjective) human expertise. Likewise, such heuristics call for compiling vulnerabilities, known problems and security issues of components into a concise risk report about the entire subsystem. By iterating this hierarchical decomposition and aggregation up to the top, we end up with a risk report that can be presented to the final decision makers for the daily business of risk control. Unfortunately, the precise process of how to aggregate risks is neither well documented nor comprehensively studied or understood (from a psychological perspective), so most of this labor is done using rules-of-thumb. The most common such rule is the "maximum principle" (cf. section 4.3.3. in BSI (2008)), 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. This problem motivated the research reported in this work, seeking to aid risk assessment and decision support by automating the previously described process, especially "approximating" the inner human decision making by using neural networks. 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.

Motivation by Example

As an abstract example, consider a simple infrastructure model composed from two representations, given as Figure 1 and Figure 2. First, we have a physical dependency model of applications on components (Figure 1), which is augmented by the logical dependency model of applications on one another (Figure 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 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 (2015)). Let CVSS(X) denote the 12-th 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. So, the risk assessment on application A would start with $CVSS(VM_1)$, $CVSS(VM_2)$. These two vectors would then go into the assessment $CVSS(AS_1)$. 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 CVSS(AS_1) only partially but not exclusively depends on CVSS(VM_1) and 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 CVSS(VM_1), CVSS(VM_2) and the information known about AS_1 into a scoring 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 CVSS(AS_1) = f(CVSS(VM_1), 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.

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 *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.

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. (2007) 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.

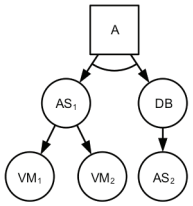

Figure 1 Dependencies of Applications on Physical Components

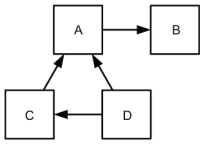

Figure 2 (Logical) Interdependencies between Applications

These challenges were independently discussed by McCalley, Fellow und Abi-Samra (1999), 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 (2012) 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, McDermott & Geer (2001), Carroll (2013), 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. (1998) or Yu et al. (2008)). 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 (2007), Ming et al (2003), or Hayden (2010) and references therein, to mention only a few. Some of these

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 Neural Network Training Examples

are specially tailored solutions (such as Ming et al (2003)) or general overviews with huge collection of heuristics and best practices; such as HEISC (2014) or Payne (2006). 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 is described by Wang (2005), who also calls for an analytical model, who 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).

The Empirical Study

We defined three scenarios inspired by real-life infrastructure security configurations, asking a set of 50 experts for their opinion about the aggregate risks, based on CVSS. The study was anonymized and delivered 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. Since our main concern is automated decision making, we leave the security-related aspects of the study and those of CVSS aside here, focusing on the NN and the risk aggregation process hereafter.

Additional to the CVSS assessment, the expert was asked to provide a total of 10 additional scores to refine the CVSS assessment. The purpose of this extension is twofold: first, it reduces the variability in the answers (as the refined scoring enforces a better thought out answer), and also provides additional insights on how experts reach their votes. Especially the latter is valuable for training the neural network subsequently.

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, Magoulas and Vrahatis (2005)) and tested a total of 13 networks, whose structure and performance results are reported in 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. For validation, we took 25% of the expert-approved test-cases and checked the aggregation result from the network against the 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.

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 be necessary for other approaches like decision trees, fuzzy logic, etc.). 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. Figure 1 and Figure 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.

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.

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 area, 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. The networks are designed to resemble human decision processes as close as possible, to the end of taking the duty of risk aggregation from the human expert. We trained and evaluated several candidate network topologies, finding that a perceptron with one hidden layer performs quite well on the risk aggregation problem, and easily integrates into standardized processes for better decision support. 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 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.

Acknowledgement

We are indebted to A. Heidorn for implementing the network training and delivering the data reported in Table 1 (Heidorn (2014)), as well as to the anonymous reviewers for providing valuable feedback and remarks.

References

Anastasiadis, A. D.; Magoulas, G. D.; and Vrahatis, M. N.; "New globally convergent training scheme based on the resilient propagation algorithm," *Neurocomputing* 64, 253-270, 2005.

Blakley, Bob; McDermott, Ellen and Geer, 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.

Bol, G.; Nakhaeizadeh, G.; Vollmer, K.-H. (Eds.); "Risk Measurement," *Econometrics and Neural Networks – Selected Articles of the 6th Econometric-Workshop in Karlsruhe*, Germany, Physica, Contributions to Economics, 1998.

Carroll, G.; "*Enterprise Compliance Today – How to aggregate risk in an Enterprise Risk Management (ERM) system*," [online] http://www.fasttrack365.com/blog/bid/347034/How-to-aggregate-risk-in-an-Enterprise-Risk-Management-ERM-system, November 2013, (accessed on November 11th, 2015).

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

Forum of Incident Response and Security Teams (first.org); "*Common Vulnerability Scoring System CVSS v3.0 Specification*," https://www.first.org/cvss/cvss-v30-specification-v1.7.pdf, 2015.

Hayden, Lance; "IT Security Metrics: A Practical Framework for Measuring Security and Protecting Data," *Mcgraw-Hill*, 2010

Heidorn, A.; „Prototypische Implementierung eines Security Risk Assessment Frameworks (SRAF) zur Erstellung und Aggregation von SRAF Graphen," Bachelor Thesis, Wernigerode: Hochschule Harz - BA, 2014.

Higher Education Information Security Council (HEISC); "Information Security Guide: Effective Practices and Solutions for Higher Education," 2014, [online] https://spaces.internet2.edu/display/2014infosecurityguide (accessed on Nov. 11th, 2015).

Kai Sun; Likhate, S.; Vittal, V.; Kolluri, V.S.; Mandal, S., "An Online Dynamic Security Assessment Scheme Using Phasor Measurements and Decision Trees," in *IEEE Transactions on Power Systems*, vol.22, no.4, pp.1935-1943, Nov. 2007.

McCalley, J.D.; Vittal, V.; Abi-Samra, N., "An overview of risk based security assessment," in *Power Engineering Society Summer Meeting*, IEEE, vol.1, no., pp.173-178 vol.1, 18-22 Jul 1999

Ming Ni; McCalley, J.D.; Vittal, V.; Tayyib, T.; "Online risk-based security assessment," in *IEEE Transactions on Power Systems*, vol.18, no.1, pp.258-265, Feb 2003

National Institute of Standards and Technology (NIST); "*Special Publication 800-30, Revision 1, Guide for Conducting Risk Assessments*," September 2012.

Payne, S.C; "*A Guide to Security Metrics*," [online] http://www.educause.edu/library/resources/guide-security-metrics, 2006 (accessed on November 11th, 2015).

Savola, R. M.; "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.

Yu, L.; Wang, S.; Lai, K.K.; Zhou, L.; "Bio-Inspired Credit Risk Analysis – Computational Intelligence with Support Vector Machines," *Springer*, 2008.