Monitoring Remote Service Platforms Using Artificial Intelligence-Based Distributed Intrusion Detection

THORSTEN WEBER[1] & RÜDIGER BUCHKREMER[2]

[1] UCAM Universidad Católica San Antonio de Murcia, Spain; e-mail: thorsten.weber@fom-net.de

[2] Institute for IT Management and Digitization, FOM University of Applied Sciences, Düsseldorf, Germany; e-mail: ruediger.buchkremer@fom-net.de

Abstract Due to their flexibility, remote support platforms are ideal for contributing to companies' digital strategy. Simultaneously, this flexibility of use cases makes it difficult to reliably detect attacks on the network infrastructure. This paper presents a proposal for the detection of fraud patterns on remote service platforms through artificial intelligence. A blockchain-based approach will be used to adapt these attack signatures to the specific use cases of remote service platform users. By employing a blockchain-based attack signature selection mechanism, remote service platform users will be able to adjust the attack signatures flexibly and in a tamper-proof manner.

Keywords: network security, intrusion detection system, remote service platform, artificial intelligence, blockchain

University of Maribor Press

DOI https://doi.org/10.18690/978-961-286-485-9.50
ISBN 978-961-286-485-9

1 Introduction

Effective plant reliability is of utmost importance for manufacturing and other industrial pursuits. Due to industrial plants' high-profile nature, unplanned downtime events can easily result in extraordinary costs (Christer & Waller, 1984). The causes of such breakdowns are numerous, and troubleshooting is typically performed by engineers or experienced technicians (Hiltunen et al., 2008). To ensure the lowest possible downtime, a company must have suitable service technicians as soon as possible on-site and available. Due to a plant's complexity, deploying an emergency service for troubleshooting can quickly turn into a planning problems (Vossing, 2017); digitization may improve the planning process's accuracy.

Remote service platforms (RSPs) are digital solutions that help companies better plan service deployment in plants. Companies can implement RSPs to train and educate workers remotely on new machines, plants, or systems. Analog monitoring processes, such as maintenance, quality assurance, and auditing, can be performed remotely, as well (Werner & Bechini, 2019). Moreover, RSPs allow remote guidance of workers and transmission of instructions. Service technicians and engineers can use RSPs to transmit real-time advice to on-site workers and repair problems from a distance without traveling. This results in less downtime and thus to a faster re-start of production after an incident.

This digitization of analog processes causes additional economic side-effects on companies. On the one hand is the direct saving of travel costs (e.g., costs for cars, flights, trains, cabs, and hotel accommodation). On the other hand, companies can redeploy their service technicians much more quickly. Service technicians must no longer "waste" time traveling and can be deployed more frequently in the same time frame. Last, saving on travel impacts a company's carbon dioxide (CO_2) footprint and can be a competitive advantage.

In summary, RSPs offer many benefits to companies. They can ensure that a service technician can quickly get to where they are needed, even if that technician might not be able or allowed to travel.

1.1 RSP Architecture

Figure illustrates a generic approach for an RSP. The architecture typically consists of three main components (Yin et al., 2006). On the one hand, it is an individual exchange and management platform to which both the service technician and the customer have access via the Hypertext Transfer Protocol Secure (HTTPS). The platform management server enables two or more participants to communicate and exchange data with each other. The management functionalities refer to access control and user management.

On the other hand, a client-side application allows users to connect to the central platform management server. Typically, these are desktop, browser, or smartphone/tablet/smart glasses applications. In a basic configuration of the communications infrastructure, two or more participants communicate via peer-to-peer (P2P)networks (Ripeanu, 2001), using the Web Real-Time Communication Protocol (Johnston et al., 2013). If a P2P connection is not feasible for technical reasons, participants switch to alternate settings (Mahy et al., 2010).

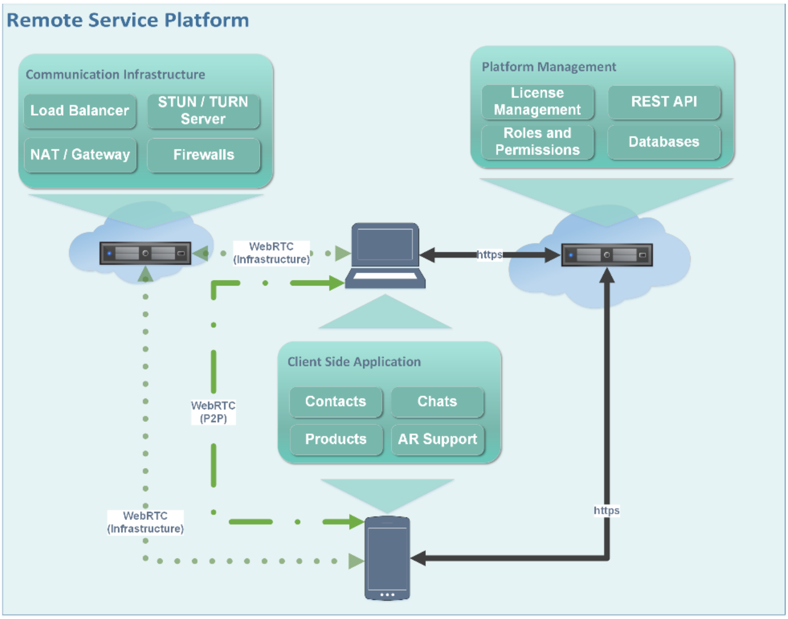


Figure 1: Typical RSP components in a nutshell.

1.2 RSP Security

An essential prerequisite for the successful implementation of RSPs is, in addition to pure functionality, confidence in the security settings of the platform's network (i.e., confidence in its security goals: confidentiality, integrity, and availability [CIA]) (Can & Sahingoz, 2015). Security assets are often critical for selecting software packages (Academy et al., 2007) and is often assumed to be naturally given (Sahay & Gupta, 2003) by a software provider.

Basis security measures of RSPs can be achieved by applying state-of-the-art security protocols, such as HTTPS or other authentication mechanisms (Kiraz, 2016). However, this basic security is not always appropriate, and advanced security mechanisms are needed. For example, the primary security mechanisms do not include protection against network-based attacks and do not allow monitoring whether a system has been exploited or tampered (Brown & Heikki, 2005; Jatti & Kishor Sontif, 2019; Liao et al., 2013). Some authors recommend implementing network intrusion detection systems (NIDSs) as the first choice for detecting network-based attacks (Debar et al., 2000; El-Bakry & Mastorakis, 2008).

The idea of intrusion detection systems (IDSs) was described in 1987 by Denning (Denning, 1987). Henceforth, the topics of IDSs were well researched by the scientific community (Khraisat et al., 2019). Today, there is a specialization trend in those systems, such as for wireless sensor networks (Can & Sahingoz, 2015), the Internet of Things (Zarpelão et al., 2017), smart grids (Jow et al., 2017), and cloud computing (Chiba et al., 2016). Specialization has the advantage that systems' unique characteristics can be considered. It is conceivable that an IDS designed for Internet of Things applications could have significantly higher requirements in terms of power consumption than, for example, an IDS developed for cloud systems.

On the other hand, many RSPs require security measures. Unauthorized platform access, attacks on communication infrastructure, and unauthorized use of premium services are only a few potential threat scenarios that could reduce confidence in RSPs. For these reasons, it is logical and consequent to develop an IDS tailored to RSPs.

2 Problem Definition

In general, the implementation of an IDS for RSPs requires attention to three main aspects. These are the mathematical requirements, the challenges for tailoring an IDS for RSPs, and the possibility of customizing and notarizing the selected configuration on the customer's part.

2.1 Mathematical Boundaries

A significant problem encountered by IDSs is the so-called base rate fallacy (Axelsson, 2000), a statistical error that may occur when determining conditional probabilities. This problem can be easily explained by applying Bayes' theorem.

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B|A)\Pr(A) + \Pr(B|\neg A)\Pr(\neg A)}$$

Assuming that 1% (Pr(¬A)) of traffic constitutes "bad traffic," such as a synchronize message flood (SYN flood), while 99% (Pr(A)) constitutes a valid connection. The IDE detection rate is 90% (Pr(B|¬A)), and the false alarm rate is 10% (Pr(B|A)).

Research question: What is the conditional probability that a connection marked by the IDS as an SYN flood is valid? What is the conditional probability that traffic is valid under the condition that the IDS triggers an alarm? Using the values mentioned above in Bayes' theorem yields the following:

$$\frac{0.10 \cdot 0.99}{10 \cdot 0.99 + 0.90 \cdot 0.01} \approx 92\%$$

Thus, if an alarm triggers the IDS, the probability is around 92% that it is a false alarm, which is an extremely high value. Ultimately, a high value can result in employees ignoring the alarm, leading to current attacks being ignored.

2.2 Tailoring Intrusion Detection to RSPs

According to (Liao et al., 2013), IDSs are divided into signature-based, anomaly-based, and specification-based systems. Signature-based and specification-based systems belong to the knowledge-based systems, while anomaly-based systems belong to the behavior-based systems. Anomaly-based IDSs detect typical user behavior and network connections; if the behavior deviates from this pattern, anomaly-based IDSs react accordingly.

Currently, there exists a trend towards specialization when developing an IDS. Research gap: a scientific approach that handles specific requirements of an IDS in the environment of RSPs is missing. The challenge for defining an IDS for RSP is the broadness of the RSP use cases, such as remote training (Masoni et al., 2017), remote audits (Teeter et al., 2010), and remote assembly (Elvezio et al., 2017). One challenge for an IDS is the ability to be adapted as flexibly as possible to existing and future RSP use cases and at the same time meet all users' data protection requirements.

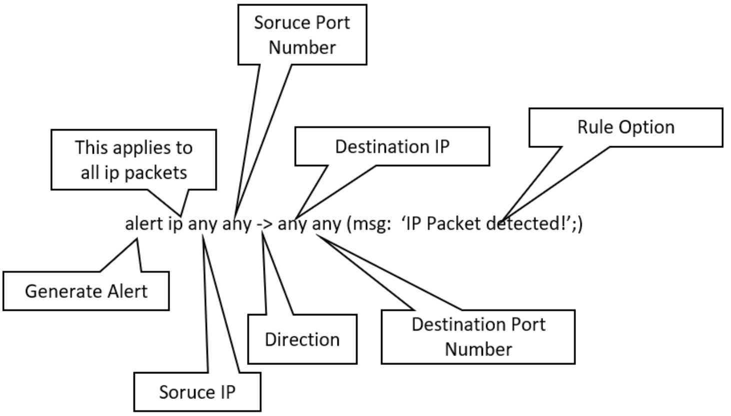
To be more precise, two artificial intelligence (AI) methods are needed. In the first step, the network traffic must be classified correctly. Using AI, received network traffic must be classified based on its properties. E.g., being HTTPS, ping, or another kind of traffic. clustering algorithms can do so (Liu et al., 2008; Münz et al., 2007). There are two approaches in principle: Supervised and Unsupervised Learning Algorithms (Sathya & Abraham, 2013). Even though their differences have been analyzed in the past, in the use case of RSPs, a priori, it is not clear which method can be used most reliably to classify the network traffic in the use cases of RSPs.

In the next step, the classified network traffic must then be analyzed and predicted whether the examined network traffic is a possible attack. The prediction of an attack can be made in various ways, for example, by analyzing the transmitted packet information using text analysis algorithms (Min et al., 2018; Stone, 2007) or using regression (Altwaijry & Algarny, 2012; Wang, 2005). Again, a priori, it is not clear which method is best suited for predicting possible attacks on RSPs. It might also be the case that a hybrid solution might be most promising.

2.3 Customization of IDSs

Typically, IDSs are configured utilizing policies (Bace & Mell, 2001). Based on the example presented in

Figure , the IDS would raise an alert containing the alert message "IP Package detected" if an IP packet from any source IP and Port would be sent to any destination IP and Port.



```
alert ip any any -> any any (msg: 'IP Packet detected!';)
```

Figure 2: IDS Policy Example for Snort ("Snort 2.1 Intrusion Detect.," 2004)

Therefore, IDS policies can determine attack patterns and read off allowed network activities. This knowledge can serve as beneficial information for an attacker to plan an attack. Securing and configuring IDS policies are therefore crucial in terms of securing infrastructures. Consequently, RSP customers are interested in confining these policies independently and need a monitoring option for selected policies and attack signatures. As such, RSP customers also need a guarantee (notarized confirmation) that the RSP provider has indeed implemented the established IDS policy and attack signatures.

2.4 Resulting Research Questions

The central question to be answered by this dissertation project is as follows*: "Is it possible to develop a privacy-compliant and customizable artificial intelligence (AI)-based attack* *detection system for remote service platforms with the highest possible detection rate and lowest possible false-positive rate, optimization of data exchange, and an intuitive visualization and reaction to detected attacks?"*

Further research questions (RQ) that this project includes are the following:

1. What are legal requirements for RSP's IDS?
2. What relevant intrusion detection system approaches already exist?
3. How can client-side applications be used to detect intrusions on RSPs?
4. How long does the learning phase of an AI-based IDS guarantee the greatest possible likelihood of attack detection?
5. How should a neural network be adjusted to distinguish between different application areas within the RSP?
6. How should the IDS react upon attack?

3 Methodology

The purpose of this chapter is to clearly outline *what* (implementation) is being done to solve each research question and *how* (means) it is being done. Moreover, this chapter addresses how the data is collected and what data can be accessed to answer the research questions.

RQs 1 and 2 serve as the basis for this dissertation, as they establish the research scope. Both research questions will be addressed via qualitative research or, to be more precise, by systematically reviewing the literature. RQ1 clarifies the legal framework in which this dissertation must operate to develop a legally secure and data- protection-compliant IDS for RSPs. The approach for solving RQs 1 and 2 is literature research, as described by vom Brocke et al. (Vom Brocke et al., 2009).

RQ 3 concerns the architecture of the software to be developed within the scope of this dissertation. The central task of the IDS is to detect attacks by examining deviations from normal behavior (Umer et al., 2017). Therefore, the IDS must receive status information of all entities being monitored. To guarantee error-free monitoring, this dissertation additionally must develop an architecture that monitors all entities reliably. Hence, RQ3 will be investigated through both qualitative and quantitative methods. A qualitative literature review must identify which IDS

architectural approaches already exist and which approaches should be considered when analyzing this research question.
On the other hand, quantitative experiments must collect and evaluate network load data and create attack signatures. An RSP typically has several connected devices, such as laptops, servers, smartphones, and smart glasses (Kao et al., 2014). By evaluating the network traffic, it is possible to check which approaches to architecture and communication with the IDS prove to be the most reliable in practice.

Since there are no reliable values for RQs 4 and 5, they must be investigated in an explorative study (Shields & Rangarjan, 2013). Therefore, a neural network will be created and trained over several periods in an attack-free test network. The attack vectors to be defined for this purpose will subsequently investigate whether the trained network recognizes attacks and how many it recognizes. Qualitative methods must be used to determine which training times are realistically achievable for actual companies (i.e., interviews with various stakeholders).

The final research question is highly individual, and it might not be possible to answer it in general terms. Instead, this dissertation aims to develop a set of recommendations based on a comprehensible presentation of various automated attack reactions. This is intended to present to users the possibilities of reacting to an attack and the consequences of these reactions.

4 Expected Results

On the one hand, this dissertation's expected results are a data protection compliant intrusion detection system that includes a set of attack signatures that are continuously improved by utilizing artificial intelligence. On the other hand, this dissertation expects to deliver a procedure allowing for the customer of the RSP to monitor the selected attack signatures and adjust them independently, if necessary.

4.1 Continuously Attack Signature Generation and Evaluation

To successfully detect an attack, the IDS must distinguish between "regular" and "attack" behaviors. Creating a continuous attack signature through log files utilization on both the client and server is, therefore, one expected outcome. Although through this dissertation, a substantial amount of actual RSP data will be available, this data will be (according to current knowledge) "attack-free." Another expected result of this dissertation is the creation of "attack" data and RSP-tailored attack signatures through penetration tests.

4.2 IDS Management via Blockchain

Besides, the goal is to develop a procedure that allows the customer of the RSP to monitor the selected attack signatures and adjust them independently, if necessary. A possible solution for this is the definition of the attack signature via blockchain. Blockchain can be used to establish a notarized definition of the selected attack signatures on the one hand and, on the other, be able to adjust the attack signature without the intervention of the platform operator.

5 Future Development

At the core of this dissertation, new attack patterns are created to detect attacks on RSPs. In particular, the data from the mobile devices that are part of the RSP will be used for this purpose. Generally, it can be assumed that client devices are mostly connected end-to-end encrypted—both with the management server and with each other for communication.

However, encrypted data packets can be examined for attack patterns to only a restricted level (Sherry et al., 2015). The data in the data packets can be analyzed for malicious content to only a limited extent. In the first step, a cloud infrastructure must be set up with which it is possible to simulate attacks on an RSP. The infrastructure must decrypt the devices' end-to-end encryption and forward the decrypted data packets to an IDS. Thus, the infrastructure must allow for the decryption of the network traffic, which must be analyzed. The decryption of encrypted network traffic can typically be achieved using a reverse proxy (Radivilova et al., 2018). In the second step, the newly built infrastructure must detect and classify new attack signatures. By performing targeted penetration tests, predefined attack patterns can be generated. Based on the performed penetration tests, the data packets analyzed by the IDS can then be stored as a new attack signature. For example, suppose a penetration test is used to conduct a brute force attack for guessing management server login data. In this case, these data packets can be uniquely recognized by the IDS and stored as a new attack signature. Later, an AI will be trained to improve the generated attack signatures continuously. Once it is possible to generate targeted attack signatures and improve them via AI, the cloud infrastructure will be connected to a blockchain. With the help of the blockchain, it should then be possible to select and monitor the various generated attack signatures in a tamper-proof manner. The result will be an IDS that specializes in RSPs, can detect attacks, and can be configured and monitored independently of the RSP operator via blockchain.

References

Academy, R. M., Studies, E., Software, E. R. P., Software, R. P., & Man-, C. M. (2007). Criteria for the selection of ERP software. Informatica Economica, XI(2), 63–66.

Altwaijry, H., & Algarny, S. (2012). Bayesian based intrusion detection system. Journal of King Saud University - Computer and Information Sciences. https://doi.org/10.1016/j.jksuci.2011.10.001

Axelsson, S. (2000). The Base-Rate Fallacy and the Difficulty of Intrusion Detection. ACM Transactions on Information and System Security. https://doi.org/10.1145/357830.357849

Bace, R., & Mell, P. (2001). NIST special publication on intrusion detection systems. In Nist Special Publication.

Brown, C. V., & Heikki, T. (2005). Information systems management handbook (8th ed.).

Can, O., & Sahingoz, O. K. (2015). A survey of intrusion detection systems in wireless sensor networks. 6th International Conference on Modeling, Simulation, and Applied Optimization, ICMSAO 2015 - Dedicated to the Memory of Late Ibrahim El-Sadek. https://doi.org/10.1109/ICMSAO.2015.7152200

Chiba, Z., Abghour, N., Moussaid, K., El Omri, A., & Rida, M. (2016). A Cooperative and Hybrid Network Intrusion Detection Framework in Cloud Computing Based on Snort and Optimized Back Propagation Neural Network. Procedia Computer Science, 83, 1200–1206. https://doi.org/10.1016/j.procs.2016.04.249

Christer, A. H., & Waller, W. M. (1984). Reducing production downtime using delay-time analysis. Journal of the Operational Research Society. https://doi.org/10.1057/jors.1984.103

Debar, H., Dacier, M., & Wespi, A. (2000). Revised taxonomy for intrusion-detection systems. Annales Des Telecommunications/Annals of Telecommunications. https://doi.org/10.1007/BF02994844

Denning, D. E. (1987). An Intrusion-Detection Model. IEEE Transactions on Software Engineering. https://doi.org/10.1109/TSE.1987.232894

El-Bakry, H. M., & Mastorakis, N. (2008). A real-time intrusion detection algorithm for network security. WSEAS Transactions on Communications, 7(12), 1222–1228.

Elvezio, C., Sukan, M., Oda, O., Feiner, S., & Tversky, B. (2017). Remote collaboration in AR and VR using virtual replicas. ACM SIGGRAPH 2017 VR Village, SIGGRAPH 2017. https://doi.org/10.1145/3089269.3089281

Hiltunen, P., Bligh, R., Klett, C., Missalla, M., & Schmidt, H. W. (2008). How to achieve high availability with large calciners and avoid unforseen downtime. TMS Light Metals.

Jatti, S. A. V., & Kishor Sontif, V. J. K. (2019). Intrusion detection systems. International Journal of Recent Technology and Engineering. https://doi.org/10.35940/ijrte.B1540.0982S1119

Johnston, A., Yoakum, J., & Singh, K. (2013). Taking on webRTC in an enterprise. IEEE Communications Magazine. https://doi.org/10.1109/MCOM.2013.6495760

Jow, J., Yang, X., & Han, W. (2017). A survey of intrusion detection systems in smart grid. International Journal of Sensor Networks. https://doi.org/10.1504/IJSNET.2017.083410

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity, 2(1). https://doi.org/10.1186/s42400-019-0038-7

Kiraz, M. S. (2016). A comprehensive meta-analysis of cryptographic security mechanisms for cloud computing. Journal of Ambient Intelligence and Humanized Computing. https://doi.org/10.1007/s12652-016-0385-0

Liao, H. J., Richard Lin, C. H., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. In Journal of Network and Computer Applications. https://doi.org/10.1016/j.jnca.2012.09.004

Liu, Y., Li, W., & Li, Y. (2008). Network Traffic Classification Using K-means Clustering. https://doi.org/10.1109/imsccs.2007.52

Mahy, R., Matthews, P., Alcatel-Lucent, & Rosenberg, J. (2010). Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN). In Internet Engineering Task Force (IETF).

Masoni, R., Ferrise, F., Bordegoni, M., Gattullo, M., Uva, A. E., Fiorentino, M., Carrabba, E., & Di Donato, M. (2017). Supporting Remote Maintenance in Industry 4.0 through Augmented Reality. Procedia Manufacturing. https://doi.org/10.1016/j.promfg.2017.07.257

Min, E., Long, J., Liu, Q., Cui, J., & Chen, W. (2018). TR-IDS: Anomaly-Based Intrusion Detection through Text-Convolutional Neural Network and Random Forest. Security and Communication Networks. https://doi.org/10.1155/2018/4943509

Münz, G., Li, S., & Carle, G. (2007). Traffic Anomaly Detection Using K-Means Clustering. GI/ITG Workshop MMBnet.

Radivilova, T., Kirichenko, L., Ageyev, D., Tawalbeh, M., & Bulakh, V. (2018). Decrypting SSL/TLS traffic for hidden threats detection. Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies, DESSERT 2018. https://doi.org/10.1109/DESSERT.2018.8409116

Ripeanu, M. (2001). Peer-to-peer architecture case study: Gnutella network. Proceedings - 1st International Conference on Peer-to-Peer Computing, P2P 2001. https://doi.org/10.1109/P2P.2001.990433

Sahay, B. S., & Gupta, A. K. (2003). Development of software selection criteria for supply chain solutions. Industrial Management and Data Systems. https://doi.org/10.1108/02635570310463429

Sathya, R., & Abraham, A. (2013). Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. International Journal of Advanced Research in Artificial Intelligence. https://doi.org/10.14569/ijarai.2013.020206

Sherry, J., Lan, C., Popa, R. A., & Ratnasamy, S. (2015). BlindBox: Deep Packet Inspection over Encrypted Traffic. Computer Communication Review. https://doi.org/10.1145/2785956.2787502

Shields, P., & Rangarjan, N. (2013). A Playbook for Research Methods: Integrating Conceptual Frameworks and Project Management. New Forums Press.

Snort 2.1 Intrusion Detection. (2004). In Snort 2.1 Intrusion Detection. https://doi.org/10.1016/b978-1-931836-04-3.x5000-0

Stone, A. (2007). Natural-language processing for intrusion detection. Computer. https://doi.org/10.1109/MC.2007.437

Teeter, R. A., Alles, M. G., & Vasarhelyi, M. A. (2010). The remote audit. Journal of Emerging Technologies in Accounting. https://doi.org/10.2308/jeta.2010.7.1.73

Vossing, M. (2017). Towards managing complexity and uncertainty in field service technician planning. Proceedings - 2017 IEEE 19th Conference on Business Informatics, CBI 2017. https://doi.org/10.1109/CBI.2017.50

Wang, Y. (2005). A multinomial logistic regression modeling approach for anomaly intrusion detection. Computers and Security. https://doi.org/10.1016/j.cose.2005.05.003

Werner, M., & Bechini, G. (2019). Customer technical support: OEM collaboration in a digitalized world. Society of Petroleum Engineers - Abu Dhabi International Petroleum Exhibition and Conference 2018, ADIPEC 2018. https://doi.org/10.2118/192623-ms

Yin, Y., Zhou, Z., Chen, Y., Liu, Q., & Long, Y. (2006). Realization of a web-based remote service platform. Proceedings - 2006 10th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2006. https://doi.org/10.1109/CSCWD.2006.253079

Zarpelão, B. B., Miani, R. S., Kawakani, C. T., & de Alvarenga, S. C. (2017). A survey of intrusion detection in Internet of Things. In Journal of Network and Computer Applications. https://doi.org/10.1016/j.jnca.2017.02.009