2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

A Distributed Multi-User Access Control Middleware for Critical Applications

Alexander Williams† *Student Member, IEEE*
Arunava Roy† *Member, IEEE*
Dipankar Dasgupta† *Fellow, IEEE*
† Center for Information Assurance, University of Memphis

***Abstract*—We present a novel access control middleware for distributed multi-user review. The system uses a fuzzy inference system trained on real world access control rules to evaluate and select reviewers as an extension to a more traditional access control system. The method is intended for high security need specific requests, as a supplement to regular access control methods. In this way, it models a multi-person access system common in mechanical controls like missile launches, bank vault opening, and other high criticality domains. The proposed method improves security by increasing the number of compromised users needed to perform an attack, taking advantage of situational awareness of peer users in a system. We evaluate the proposed system with an example implementation based on a real-world organization, and show that the system can be used to effectively implement a secure resource access control system. Our work contributes to the growing body of research into fuzzy-logic access control, ML in access control, and multi-user authentication systems.**

***Index Terms*—Access Control, Fuzzy Systems, Security**

I. INTRODUCTION

Access control remains pivotal in the security management of diverse sectors like healthcare, finance, and government agencies. Notwithstanding the advancements in automated access control methods and machine learning models assessing system threats [1], challenges persist. Some organizations have unique flexibility demands inadequately served by existing models, or data security needs resistant to automation. Numerous systems addressing these issues have recently been proposed, including Bayesian and probabilistic methods, and machine learning based strategies.

Existing access control models predominantly employ role or task-based classifications to grant or deny resource access. However, these models are insufficient in several ways. Notably, controlling resources that require high granularity, such as medical records or compartmentalized information, is challenging. Existing systems can manage this granularity, but at the cost of an expansive rule corpus, straining specific role/group hierarchies or attribute-based methods [2].

Additionally, these kinds of rigid rules based access control schemes lack inherent flexibility when facing a changing and dynamic topology in the underlying access control system. As accounts get compromised, actors turn from good to bad, or resource access needs change. Moreover, insider threats significantly jeopardize traditional systems. Previous trusted users or external threats using stolen credentials can cause substantial damage with elevated access rights [3]. Some proposed solutions detect anomalous behavior, but this alone is insufficient.

To address these challenges, we introduce the Distributed Multi-User for Review Access Control Systems (DMURACS), a middleware access control system designed for high-security and high-risk applications. DMURACS is not in and of itself an access control system. Instead, when paired with a traditional RBAC/ABAC or similar system, it helps control access to critical security or safety sensitive resources and implements a human reviewer based system as a check on anomalous or high risk requests.

In DMURACS, when the underlying system flags a request as being high risk or needing additional review, a trained fuzzy system identifies potential reviewers using specific criteria and sends them a request. Access is granted upon consensus or voting by the reviewers. This builds upon prior research [4], but offers improved implementability and computational complexity. In this way, DMURACS acts as a Multi-Factor Authentication methodology, but instead of verifying the identity of a user, it adds checks to the appropriateness of a resource request. Two-person systems, modernized to multi-person systems in our case, have proven effective in controlling critical defense, security, and financial systems access [5]. The DMURACS system leverages this proven concept for a broader range of applications.

DMURACS operates in two parts: a Self-Organizing Map (SOM) trained on access requests, followed by fuzzy rule extraction from the trained model. This approach enables greater decision explainability and system parameter modification or rule addition as needed.

DMURACS offers these key benefits:

- Facilitates direct human intervention in high-risk resource access control situations.
- Increases security: a malicious agent needs to compromise multiple accounts, not knowing in advance who the reviewers will be.
- Enhances system transparency: the fuzzy rule-based reviewer selection and human-justifiable approvals or denials make audit trails clear and comprehensive.

This work is partially supported by NCAE-C Cyber Research Innovation grant (NCAE-C-002-2021) to the University of Memphis. We would acknowledge the contributions of Dr. Mike Nolen, Ryan Wickman, Cody Lightfoot and others who worked on various aspects of the project at different times.

The remainder of the paper is organized as follows: Section 2 reviews access control literature and attempts to solve the stated challenges. Section 3 details the DMURACS system, including architecture, SOM training, fuzzy rule extraction, and feedback loop. Section 4 offers a proof of concept using an anonymized real-world organizational structure. Section 5 concludes and suggests future research directions.

II. Related Work

Throughout recent years, a multitude of access control methodologies have been proposed, from early mandatory access control systems and ACLs, to attribute, roll, and task based access controls. These methods vary in applicability to various domains, applications, and organizational structure, with trade-offs for flexibility, security, and manageability.

The manual implementation and administration of these systems can become quite unweildly, so recent research has turned to machine learning for the creation, verification, and administration of these control systems. A thorough review of modern machine learning approaches to access control was put together by Nobi et. al [1], showing the expanse of work in this area. However, addressing evolving systems and the threats to those systems is still an under-explored area of research, and traditional systems either do not address changes to system architecture while running, or do so only inadequately.

In order to mitigate these limitations, and to provide means by which a system can evolve over time, various groups have developed and introduced Risk Based Access Control.These are typically extensions to RBAC/ABAC that dynamically evaluate the relative risk of a request, and make access control decisions either in whole or in part based on those evaluations [6]. Various approaches have been described in recent literature, and a brief, non-exhaustive overview of this literature is provided in **Table I**.

As can be seen, many of these approaches rely on machine learning and neural networks for their advancements. Our work with DMURACS fits into this space, advancing the space of helping access control systems evolve and adapt to a changing threat landscape. In particular, we turn to fuzzy inference systems, which are a natural fit to risk based access control modalities as risk is rarely a true or false proposition and must be interpreted heuristically [7]. A fuzzy based approach has been evaluated favorably in contexts such as cloud computing [8] and medical information security [9].

III. DMURAC: Design and Implementation

A. Overview and Motivation

Much research focuses on external security threats like DDOS attacks and hacking. Yet, a major, often ignored, threat source is internal users inappropriately accessing or removing information. The 2022 Verizon Data Breach Investigation Report indicates 82% of data breaches involved a human element, mostly external. In healthcare, however, 39% of breaches came from internal actors with legitimate system access. [15]

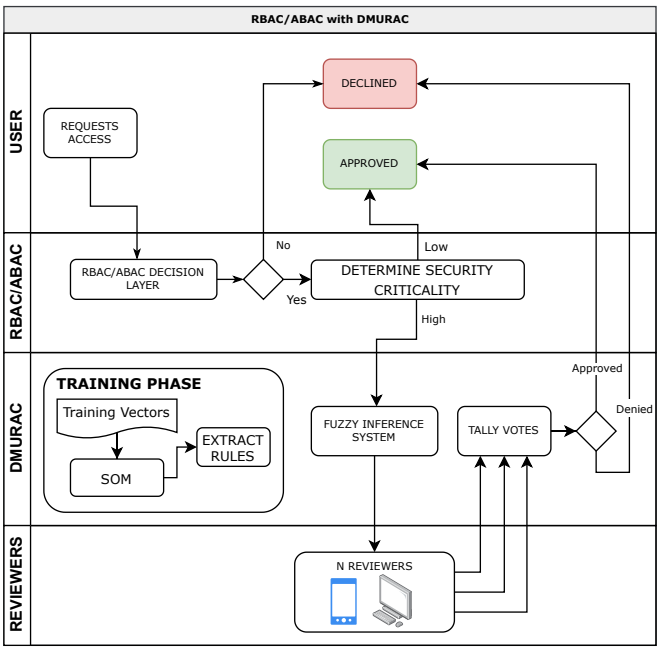

Fig. 1: DMURACS flowchart including training phase and request processing

The challenge in preventing such incidents is the broad access internal users often have, leading to potential misuse. Traditional access control systems grant rights based on roles, attributes, or tasks, with limited granularity. Implementing more detailed access control manually is complex and difficult to verify correctness as updates need to be made when staffing is changed.

In a variety of high risk or safety critical applications, organizations turn to two-person control systems. These include requiring multiple authorizations for applications like nuclear weapons launch, large financial transactions, and pharmaceutical quality assurance. These systems largely stem from an effort to reduce the impact of human compromise, either malicious or accidental.

Clearly, such a system, if used each time an access request was made would be prohibitively expensive in scale and time, computational and human. Therefore, our approach is intended to be an extension of a more traditional access control system, where access to resources that are particularly critical or high risk can undergo further review by peers, who would be familiar with the operational status and needs of a user. This mimics the use by other safety critical applications. A bank does not require two people to sign off every time someone enters the breakroom with their badge, but opening a vault or safe, or performing a very large financial transaction, likely would require more than one person's involvement. There would be some trade-off in terms of speed and availability of resources. In the digital age, we are accustomed to access control systems being responsive and immediately available at all times. Alternatively, historically when resources require additional guarding or management, a slow manual process usually through a central administrator

TABLE I: Recent Works Addressing Evolving Systems and Threats

Authors	Summary	Limitations
Ashfar et al. [10]	Continuously evaluates a system using ML for changes in employee behavior, denying access when it detects anomalies	Possibility of false positives during rapid changes in organization, emergencies, or other dynamic situations
You et al. [11]	Addresses asymmetry in accept/deny data for continuous training using batch learning and minority class boosting	Updates to system and responding to breaches still requires going through a retraining cycle, slowing response to improper access
Rubio-Medrano et al. [12]	Developed formal framework for assessing ABAC systems for vulnerabilities an faults, especially to attribute forging attacks	Does not address other kinds of attacks, including insider attacks from accounts with appropriate attributes
Srivastava et al. [13]	Proposes Risk Adaptive Access Control to assess genuineness of a request and the risk of granting access before deciding on access.	Heavily leverages hidden layer neural network, limiting explainability of system, may not adhere to auditing requirements for critical systems
Fragkos et al. [14]	Integrates Bayesian based user trust evaluation into RBAC.	Probabilistic method may depend on a previously trusted account breaching multiple times. Can be slow to respond to changes.

is performed. Our solution strikes the balance, providing manual review, while still providing a digital interface and higher up-time than a single administrator system would.

Importantly, this system does not have to rely on traditional hierarchies for access control decisions. While a supervisor or other kind of dedicated resource controller would likely be included in the set of appropriate reviewers, coworkers familiar with the responsibilities and roles of the requester also provide checks and balances to the access of other users. This responsibility dilution prevents targeted attacks on specific users from compromising entire systems, limits the amount of damage that can be done by a single attacker, and introduces human accountability into a system.

Our proposed design takes cues from this approach and implements a shared decision making and risk assessment model into modern access control. It does this by requiring the approval of multiple users in a system to ensure that responsibility is shared and to hopefully minimize the likelihood of malicious access. We describe a simple strategy for identifying appropriate reviewers in an organizational agnostic way. This system then, when access to a critical resource is requested, automatically determines a pool of appropriate reviewers and selects from this pool. It then requests a review of the access request, for example, with a push notification to a phone or computer. This method benefits from user familiarity with other 2-factor authentication methods like Duo, and provides similar security benefits.

The novelty in our system lies in the overall procedure of multi-user approval for access control requests, as well as the selection algorithm and implementation. The system as described is highly adaptable to a variety of organizations and can be readily integrated with other, more traditional access control systems if desired.

B. Initial Setup

A DMURACS system is straightforward in setup and operation. **Figure 1** shows an overview of this system. Initial set up of DMURACS occurs after a traditional RBAC/ABAC system is implemented, and attributes of users, roles, and resources in the system are well classified.

Initializing the Fuzzy Inference System at the core of DMURACS involves training a SOM on vectors that contain the user attributes of a requesting user, a potential reviewer, the resource being requested, and an output attribute, which is the target confidence level of the FIS in how applicable the reviewer is for reviewing the user's request for the resource. The SOM is then trained on these vectors, and the rules are extracted using the process outlined in **Section IV**.

C. Execution and Evaluation

When a user requests a resoure, first the request is passed through a traditional access control system, defining whether or not the user should have access to the resource. If so, and if the system flags the resource as being of particularly high security or safety risk, then it passes the information to DMURACS, which uses the vectors for the requesting user and the resource and searches through applicable users, rating their fitness for reviewing the request, and sending the request notification to a pre-set number of users above a threshold. If enough "approve" votes are received, the requester is granted access. All actions are logged, including the reviewer selection, voting results, and any subsequent vector updates. To prevent reviewer fatigue and potential abuse, the selection odds for recent reviewers are temporarily reduced.

System auditing allows for tweaks in fuzzy system variables if necessary, offering transparency compared to "black box" systems. Forensic audits are easily performed due to the traceable decision-making process, enhancing the system's explainability.

IV. SOM Rule Inference and Extraction

Kohonen Self Organizing Maps (SOMs) [16] are well explored technology, with significant applications in a wide variety of fields, so many groups have developed tools for extracting inferences and data from these. Here though, we iterate on a model used to classify faults in electrical transformer production, where a SOM is used to generate 2-dimensional projections of higher dimensional fault information. The rules inferred by the SOM are transformed into fuzzy rules. [17], [18].

Algorithm 1: Kohonen SOM with Rule Extraction

```
   input : Training Vector X
1  Randomize node weights
2  let rules = []
3  for x_n ∈ X do
4  |  for w_i ∈ W do
5  |  |  Let B_x = arg min_{i∈I} ||x − W_i||^2
6  |  end
7  |  w_i(t+1) = w_i(t) + σ(t)α(t)(x_n − w_i)
8  end
9  for cluster i do
10 |  C_i = arg max_{w∈W_i} { Σ_{n=1}^{N_i} e^{−||w_i − x_n||} }
11 end
12 for C_i ∈ {C} do
13 |  rule = []
14 |  for x_n ∈ x : x in input to cluster do
15 |  |  m = min x_n
16 |  |  M = max x_n
17 |  |  c = w_in
18 |  |  A_in = exp(−1/2((x_n − x)/‖M − n‖))
   |  |   rule.append(x_n ∈ A_in)
19 |  end
20 |  rules.append(rule)
21 end
   output: rules
```

The general steps to performing this analysis are as follows:

1) Creation and training of SOM
2) Find clusters in the trained model and determine ideal representative node
3) Create membership functions using ideal nodes
4) Combine membership functions and SOM cluster labels into fuzzy rules

A. Training SOM and Identifying Clusters

The training and set up of SOMs are well known to the field at this point, so further exploration is left for the reader, but once trained, we are left with a lower dimensional clustering of data points. It is these clusters that we are primarily interested in, as heuristically, they present a collection of vectors in the training data that roughly correlate with our target outcome of reviewer quality. From these clusters, we can derive rules for our fuzzy system. This initialization is represented by lines 1-8 in **Algorithm 1**.

Next, we identify the clusters in the SOM hyper-grid. A variety of ways of performing this task have been proposed throughout the years, including boundary drawing algorithms in [19], C-means or K-means clustering as in [20], [21], or others. We choose a modification of a process described in [17] called subtractive clustering. For all nodes in a given cluster, we find the maximum of a likelihood function for how good of a center point an individual node is:

$$C_i = \arg\max_{w \in W_i} \left\{ \sum_{n=1}^{N_i} e^{-\|\vec{w}_i - \vec{x}_n\|} \right\} \tag{1}$$

where N_i is the number of vectors in the training set that pointed to the current cluster.

B. Fuzzy Inference System Rule Extraction

The literature on fuzzy inference systems is, at this point, well developed, with many excellent resources describing the properties and creation of such systems [22], [23]. Therefore, we will just draw attention to some specifics required for rule extraction.

A fuzzy rule is an IF-THEN statement establishing the membership of variables from the input vector to degrees of membership in the output function. These rules are of the form:

$$R_i : \textbf{IF } x_0 \in A_{i0} \textbf{ AND} \cdots \textbf{AND } x_n \in A_{in} \Rightarrow y_i = i \tag{2}$$

Where each A_{in} represents a membership function for the i-th membership class for the n-th member of the input vector.

Each rule, once evaluated, will have a weight ϕ_j, which is defined as some AND operator over the membership functions evaluated at the respective input. While any AND style operator will work, we choose a simple product operator.

$$\phi_j = A_{i1}(x_1) \otimes \cdots \otimes A_{in}(x_n) \tag{3}$$

Finally, the fuzzy inference system produces an output. There are, again, several popular output methodologies but we will use the most common: center of mass.

$$F = \frac{\sum_{j=1}^{n} \phi_j y_j}{\sum_{j=1}^{n} \phi_j} \tag{4}$$

The preliminaries addressed, we turn to extracting rules from our SOM, represented by lines 12-20 in **Algorithm 1**. Recall that our last step yielded a set of cluster centers $\{C_i\}_{i=1}^{n}$ that were best representations of their cluster. Each of these clusters now represent a fuzzy rule. We then construct a Gaussian membership function using the following values extracted from the SOM: $\boldsymbol{c}$: weight from input x_n and the cluster center node, $\boldsymbol{m}, \boldsymbol{M}$: minimum and maximum values respectively of x_n pointing to the cluster i. We combine these into a membership function:

$$A_{in} = \exp\left(-\frac{1}{2} \left(\frac{x_n - c}{\|M - m\|} \right) \right) \tag{5}$$

This provides a Gaussian curve where closer relationship to the cluster center yields stronger activation of the relevant rule. This provides an intuitive heuristic mapping from hard values to cluster membership. If "input values" are "around c" then the vector belongs to cluster i. The closer the input values match c, the more confident the system is that the vector belongs to the cluster, and so it will have a higher firing weight.

The relevant values can be directly extracted from the SOM model and imported into an off-the-shelf FIS, from MATLAB, scikit-fuzzy, or other related system. A code sample in MATLAB is available in the supplementary material, demonstrating a concrete application.

V. Experimental Implementation

To evaluate the effectiveness of the proposed multi-user access control system, we modelled a real-life local company and their resource access request needs, including their employees and the hierarchical relationships between them. The company has several departments, each with its own set of resources, access policies, and overall structure. We consulted with the company's information security personnel to create a set of access request scenarios that represented their typical access request patterns, as well as requests that are abnormal and should be subject to higher scrutiny. These scenarios were used to generate the training vectors for the self-organizing map (SOM).

The training vectors were created in consultation with information security personnel from the organization, who provided us with a list of access request scenarios that were representative of the types of requests that the organization received. These scenarios included requests for access to specific files, databases, and applications, as well as requests for changes to access policies and other administrative tasks. The chosen input variables are listed in **Table II**.

TABLE II: Training Vector

Name	Meaning
dist_ur	Distance on org chart between the requester and reviewer
cu_clr_lvl	Clearance Level of User (1-5)
u_area	The physical area the requesting user is (supposed) to be working in
r_has_access	Does reviewer have access to the resource themselves?
dist_doc	min distance on org chart to any owner of the document
d_sec_level	security level of resource (1-5)
d_area	Physical location of resource
output	should reviewer be in pool?

We used MATLAB's Neural Network Toolbox to train the SOM using the training vectors. The SOM was trained using unsupervised learning with a learning rate of 0.5 and a neighborhood distance of 2. We used a normalization function to scale the input data before training the SOM. The SOM had a 10x10 grid of neurons and was trained for 100 epochs.

After training the SOM, we visualized the neuron activations using a SOM hit-map and a U-matrix. The hit-map showed the distribution of input patterns across the SOM, while the U-matrix showed the average distance between neighboring neurons. The visualizations helped us identify clusters of neurons that corresponded to groups of similar access requests.

We then used subtractive clustering to identify the output clusters, then extracted rules from the SOM clusters in the manner described in section 4. These extracted rules were then loaded into MATLAB's FIS for evaluation, and into Python 3.4 and Scikit-learn for implementation of an authentication server. To test the system, we implemented a prototype server and mobile application to evaluate the workflow and experimentally observe correct reviewer selection. The mobile application was loaded on to multiple Android phones, and sample requests were made.

The server was created using NodeJS, with Python 3.4 and Scikit-Learn with the SciKit-Fuzzy plugin being used to implement the fuzzy inference system and evaluate user requests. Organizational structure, user, and document databases were stored in MongoDB, which allowed for easy storage of hierarchical structures in a JSON format. This database was formed in consultation with IT security from a local, real-world company, and features 250 individual users in a hierarchical structure with multiple levels of sub-department, along with 100 safety or security sensitive resources like employee HR files and heavy machinery. This structure allowed us to easily parse and simulate real world conditions for a typical use case for our system. Screenshots and code for the systems available in supplementary material.

In order to test this prototype implementation, we simulated 6000 requests for access from a subset of 150 employees of our real-world company, requesting access to 50 different resources. Each request polled the database of users for potential reviewers and formulated a list of reviewers that satisfied a cutoff of 75% likelihood of being a good reviewer. It is important to note that all the requests were for resources that would be considered "reasonable" from a role perspective at the organization, but that due to the increased risk of access, should be reviewed before accepting.

We found that in our system, the vast majority, 98.7%, of requests, were able to find a sufficiently large group of acceptable reviewers for requests. Acceptability was defined here as meeting the 0.75 confidence threshold by the FIS. Where the system struggled in this task were primarily high level employees requesting access to documents with very few people in the circle of acceptable reviewers. This is consistent with expectations, that there would be very few people in an organization who could review a VP's request for high level document access. Further work is needed to address these edge cases.

VI. Conclusion

In this work we introduced DMURACS, a novel access control middleware for access control systems that intervenes in the approval process for high security or sensitivity resources. It improves upon security by introducing two-person (or more) approval processes that provides an additional review method for these high risk requests. Instead of direct managerial approval, a group of users, which can be peers, supervisors, or anyone familiar with the specific requester and resource, provides a sanity check and additional layer of security on a request. It does this with a fuzzy inference sys-

tem, the rules of which are extracted from a SOM's weights and vectors. Selected reviewers are sent push notifications to approve or deny requests.

We evaluated DMURAC by modelling a local, real-world organization in consultation with IT security staff from the organization to create the training data for the fuzzy inference system. The model was then created and deployed, and trial requests performed. We showed that the system performs successfully in this limited trial run.

Future directions can extend the implementation by integrating it with existing systems, using our system only when an existing implementation is unsure or denies access. Additionally, case studies on full deployments to large scale organizations would prove useful in determining the human characteristics and limitations of our approach.

REFERENCES

[1] M. N. Nobi, M. Gupta, L. Praharaj, M. Abdelsalam, R. Krishnan, and R. Sandhu, "Machine Learning in Access Control: A Taxonomy and Survey," Jul. 2022, arXiv:2207.01739 [cs]. [Online]. Available: http://arxiv.org/abs/2207.01739

[2] T. Tsegaye and S. Flowerday, "A Clark-Wilson and ANSI role-based access control model," *Information & Computer Security*, vol. 28, no. 3, pp. 373–395, Jan. 2020, publisher: Emerald Publishing Limited. [Online]. Available: https://doi.org/10.1108/ICS-08-2019-0100

[3] S. L. Garfinkel, N. Beebe, L. Liu, and M. Maasberg, "Detecting threatening insiders with lightweight media forensics," in *2013 IEEE International Conference on Technologies for Homeland Security (HST)*, Nov. 2013, pp. 86–92.

[4] D. Dasgupta, A. Roy, and D. Ghosh, "Multi-user permission strategy to access sensitive information," *Information Sciences*, vol. 423, pp. 24–49, Jan. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025516312129

[5] R. Pedersen, "Two-Person Control: A Brief History and Modern Industry Practices," Tech. Rep. SAND2017–7995, 1374246, 655768, Jul. 2017. [Online]. Available: https://www.osti.gov/servlets/purl/1374246/

[6] E. Celikel, M. Kantarcioglu, B. Thuraisingham, and E. Bertino, "A risk management approach to RBAC," *Risk and Decision Analysis*, vol. 1, no. 1, pp. 21–33, Jan. 2009, publisher: IOS Press. [Online]. Available: https://content.iospress.com/articles/risk-and-decision-analysis/rda02

[7] D. Choi, D. Kim, and S. Park, "A Framework for Context Sensitive Risk-Based Access Control in Medical Information Systems," *Computational and Mathematical Methods in Medicine*, vol. 2015, p. e265132, May 2015, publisher: Hindawi. [Online]. Available: https://www.hindawi.com/journals/cmmm/2015/265132/

[8] A. Chen, H. Xing, K. She, and G. Duan, "A Dynamic Risk-Based Access Control Model for Cloud Computing," in *2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom)*, Oct. 2016, pp. 579–584.

[9] J. Li, Y. Bai, and N. Zaman, "A Fuzzy Modeling Approach for Risk-Based Access Control in eHealth Cloud," in *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, Jul. 2013, pp. 17–23, iSSN: 2324-9013.

[10] M. Afshar, S. Samet, and H. Usefi, "Incorporating Behavior in Attribute Based Access Control Model Using Machine Learning," in *2021 IEEE International Systems Conference (SysCon)*. Vancouver, BC, Canada: IEEE, Apr. 2021, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/9447115/

[11] M. You, J. Yin, H. Wang, J. Cao, and Y. Miao, "A Minority Class Boosted Framework for Adaptive Access Control Decision-Making," in *Web Information Systems Engineering – WISE 2021*, ser. Lecture Notes in Computer Science, W. Zhang, L. Zou, Z. Maamar, and L. Chen, Eds. Cham: Springer International Publishing, 2021, pp. 143–157.

[12] C. E. Rubio-Medrano, L. Claramunt, S. Jogani, and G.-J. Ahn, "Proactive Risk Assessment for Preventing Attribute-Forgery Attacks to ABAC Policies," in *Proceedings of the 25th ACM Symposium on Access Control Models and Technologies*, ser. SACMAT '20. New York, NY, USA: Association for Computing Machinery, Jun. 2020, pp. 131–144. [Online]. Available: https://doi.org/10.1145/3381991.3395615

[13] K. Srivastava and N. Shekokar, "Machine Learning Based Risk-Adaptive Access Control System to Identify Genuineness of the Requester," in *Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough: Latest Trends in AI*, ser. Studies in Computational Intelligence, V. K. Gunjan, J. M. Zurada, B. Raman, and G. R. Gangadharan, Eds. Cham: Springer International Publishing, 2020, pp. 129–143.

[14] G. Fragkos, J. Johnson, and E. E. Tsiropoulou, "Dynamic Role-Based Access Control Policy for Smart Grid Applications: An Offline Deep Reinforcement Learning Approach," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 4, pp. 761–773, Aug. 2022, conference Name: IEEE Transactions on Human-Machine Systems.

[15] Verizon, "2022 Data Breach Investigation Report." [Online]. Available: https://www.verizon.com/business/resources/Td6c/reports/dbir/2022-data-breach-investigations-report-dbir.pdf

[16] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, Jan. 1982. [Online]. Available: https://doi.org/10.1007/BF00337288

[17] R. Naresh, V. Sharma, and M. Vashisth, "An Integrated Neural Fuzzy Approach for Fault Diagnosis of Transformers," *IEEE Transactions on Power Delivery*, vol. 23, no. 4, pp. 2017–2024, Oct. 2008, conference Name: IEEE Transactions on Power Delivery.

[18] A. C. M. da Silva, A. R. Garcez Castro, and V. Miranda, "Transformer failure diagnosis by means of fuzzy rules extracted from Kohonen Self-Organizing Map," *International Journal of Electrical Power & Energy Systems*, vol. 43, no. 1, pp. 1034–1042, Dec. 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S014206151200289X

[19] J. Malone, K. McGarry, S. Wermter, and C. Bowerman, "Data mining using rule extraction from Kohonen self-organising maps," *Neural Computing & Applications*, vol. 15, no. 1, pp. 9–17, Mar. 2006. [Online]. Available: https://doi.org/10.1007/s00521-005-0002-1

[20] A. Gosain and S. Dahiya, "An effective fuzzy clustering algorithm with outlier identification feature," *Journal of Intelligent & Fuzzy Systems*, vol. 41, no. 1, pp. 2417–2428, Jan. 2021, publisher: IOS Press. [Online]. Available: https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs201858

[21] D. A. Linkens and M.-Y. Chen, "Input selection and partition validation for fuzzy modelling using neural network," *Fuzzy Sets and Systems*, vol. 107, no. 3, pp. 299–308, Nov. 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165011497003229

[22] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985, conference Name: IEEE Transactions on Systems, Man, and Cybernetics.

[23] J. Yan, M. Ryan, and J. Power, *Using fuzzy logic: towards intelligent systems*. USA: Prentice-Hall, Inc., Jan. 1995.