# Machine Learning in Access Control: A Taxonomy

[Systematization of Knowledge Paper]

Anonymous Author(s)

## ABSTRACT

Developing and managing access control systems is challenging due to the dynamic nature of users, resources, and environments. Recent advancements in machine learning (ML) offer promising solutions for automating the extraction of access control attributes, policy mining, verification, and decision-making. Despite these advancements, the application of ML in access control remains fragmented, resulting in an incomplete understanding of best practices. This work aims to systematize the use of ML in access control by identifying key components where ML can address various access control challenges. We propose a novel taxonomy of ML applications within this domain, highlighting current limitations such as the scarcity of public real-world datasets, the complexities of administering ML-based systems, and the opacity of ML model decisions. Additionally, we outline potential future research directions to guide both new and experienced researchers in effectively integrating ML into access control practices.

## CCS CONCEPTS

• **Security and privacy** → **Access control**; • **Computing methodologies** → **Machine learning**.

**ACM Reference Format:**

Anonymous Author(s). 2022. Machine Learning in Access Control: A Taxonomy: [Systematization of Knowledge Paper]. *J. ACM* 37, 4, Article 111 (August 2022), 12 pages. https://doi.org/10.1145/1122445.1122456

## 1 INTRODUCTION

Researchers have shown significant interest in applying machine learning (ML) to various aspects of access control, such as policy mining, verification, monitoring, and administration. Although still in its infancy, ML applications in access control are evolving with the goal of replacing language-based security policies with trained ML agents. Figure 1 provides an overview of access control systems and their pipelines. Traditional systems involve roles, attributes, policy engineering, verification, and administration, while ML-based systems replace security policies with trained ML agents, which involve training, verification, and administration of the agent [79]. Optional components are indicated by dashed boxes in Figure 1.

ML has proven highly successful in solving complex problems across various domains, outperforming manual human-driven processes. In access control, manual solutions often fail to achieve optimal performance and may grant unauthorized access. Consequently, there are significant opportunities for improvement in access control. Researchers have leveraged ML to develop more efficient solutions, such as providing decisions for unseen access scenarios [98], automating laborious tasks like attribute extraction from natural language [3–5, 46, 75], mapping roles and permissions [78, 101], extracting security rules from access logs [30, 60, 73], and deriving access control policies from user stories [46]. ML has also been used for access policy verification [47, 49] and monitoring suspicious activities [96]. Additionally, researchers have proposed using ML for access control decision-making, where trained models determine whether access requests should be granted or denied [22, 25, 59, 67, 79].

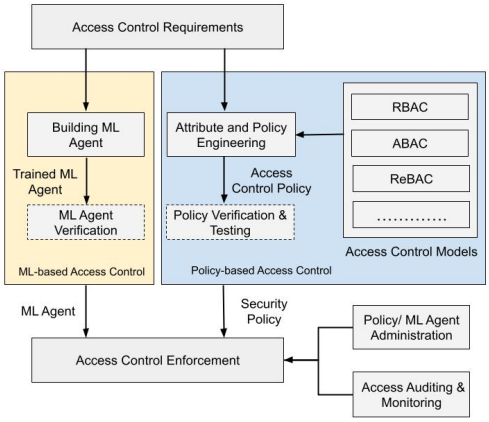


**Figure 1: Access Control Landscape reflecting both Policy and Machine learning based enforcement.**

Evidently, the rapid emergence and utilization of ML has shown a significant potential in improving and reshaping the field of access control. However, several challenges need to be addressed. A major obstacle is that researchers tend to apply ML methods on a case-by-case basis [4, 15, 30, 39, 61, 96], and thereby, there is no common strategy for using ML in the access control domain. This leads to a lack of in-depth insights and the absence of a holistic view of the application of ML in access control. In addition, there is a lack of research efforts that address the best strategy to determine the most effective ML technique given a particular access control problem. Another limitation pertains to the availability of data. There is a noticeable lack of quality datasets from the real-world organization [37, 47]. Even though some public datasets are available, they are typically anonymized datasets which, in most cases, exclude relevant information necessary for expressing a complete access control state of the system [74]. Considering all the aforementioned limitations, it is essential to have a holistic view of the utilization of machine learning for access control, which, in turn, will help to



0004-5411/2022/8-ART111 $15.00

https://doi.org/10.1145/1122445.1122456

shed light on the underdeveloped areas and determine the future directions in the domain.

In this paper, we perform a detailed review and summarize existing literature that uses ML to solve different access control problems. To the best of our knowledge, this is the first comprehensive work that offers an encyclopedic view towards outlining the application of machine learning for access control. Our systematization makes the following contributions.

- We comprehensively review existing access control literature that uses machine learning and discuss various research works done in different sub-domains of access control.
- We propose a novel taxonomy of machine learning in access control, and highlight research at each stage as the domain has evolved chronologically.
- We summarize the publicly available real-world datasets used for machine learning based access control research.
- We highlight open challenges and limitations faced by the research community, as well as provide future research directions to thrive in this critical security domain.

# 2 ML IN ACCESS CONTROL TAXONOMY AND SCOPE

This paper dives into the exciting world of machine learning in the access control pipeline. We've crafted an abstract taxonomy to showcase the intricate relationships among various components and building blocks of access control research, helping to organize our work, as illustrated in Figure 2.

We've categorized the available work into two major areas: ML Assisted Access Control' and ML Based Access Control'. The first category includes innovative literature that enhances traditional policy or role-based access control processes, such as attribute engineering, policy mining/extraction, role/permission assignments, and policy verification. Overall, those methods mostly deal with processes in ABAC, RBAC, ReBAC. The second category is even more groundbreaking, featuring articles that *replace* traditional access control policies with an ML model. In this scenario, the model itself acts as the policy, making access decisions autonomously.

Throughout the paper, we explain each of the branches of this figure, intending to solve some of the key questions as follows. These questions are answered in Section 3 and 4 for each work discussed, as well as summarized per-subdomain in Tables 1-5.

(1) What are the target access control models? In particular, is the proposed ML approach applicable for the access control domain as a whole or only suitable for any particular model?
(2) What are the ML methods that the respective approach uses?
(3) Why does the respective approach use ML, and to what extent ML method contribute?
(4) What are the input and output of the ML model? Can the trained machine learning model make access control decisions, or does the corresponding method only use ML to improve or automate access control sub-processes?
(5) What kind of data was used for training the ML algorithm?

We also explore any possible enhancements related to ML in access control, such as adversarial attack, explainability, bias, etc.

## 2.1 Corpus Collection

This systematization is based on published research from 2006 to the end of 2024. Before 2006, no ML-based access control solutions were found. We reviewed papers from various sources, including Google Scholar, ACM Digital Library, IEEEXplore, Springer, and preprints from arXiv.

We chose to read individual papers rather than rely on keyword searches in databases, as this approach ensured no relevant articles were overlooked. The terminology used to describe ML practices in access control is highly varied due to the field's nascent nature. Consequently, keyword searches might have excluded significant literature. By manually reading and selecting papers, we compiled a comprehensive and contextually relevant collection for our study.

We also summarized datasets used in these studies to provide an overview of available datasets, their applications, and limitations. All publicly available datasets are summarized in Table 1.

## 2.2 Timeline

We present the timeline of seminal works using ML for access control in Figure 3. We sort the timeline according to the published year and illustrate how the application of ML evolved in the access control domain. As summarized in Figure 3, ML in access control is fairly new introducing its concept first in 2006. However, the application of ML in access control domain is emerging fast, and researchers published most of the work in recent years.

# 3 ML ASSISTED ACCESS CONTROL

This section provides an overview of literature that leverages machine learning to address issues related to traditional access control models. In these studies, machine learning isn't used to directly predict access decisions. Instead, it optimizes various processes such as policy mining, role mining, and rule mining.

## 3.1 Policy Mining

High-level access control models like ABAC and ReBAC are favored for their flexible policies and reduced management burdens, supporting dynamic and complex security policies. However, transitioning from lower-level policies like ACLs is challenging. Policy mining techniques use user and resource attributes, their values, and the system's current access control state as input. For ABAC and ReBAC, algorithms generate rules that grant the same permissions [18, 20]. In contrast, RBAC mining algorithms produce permission-to-role (PA) and user-to-role (UA) assignments [78, 101]. Table 2 reports access control policy mining approaches using ML.

### *3.1.1 Attribute Based Access Control (ABAC).*

**Attributes and Policy Extraction from Natural Language.** Natural language policies, being the preferred expression of policy [3], need to be transformed into a machine-readable form. Several researchers attempted to process such policies to extract access control-related information, including identifying policy sentences, triples of subject-object-action, etc. While manual extraction of such information is inefficient as the task becomes repetitive, requires more time, and is error-prone, several other approaches have been proposed to automate the process [75, 76, 87, 88, 97].

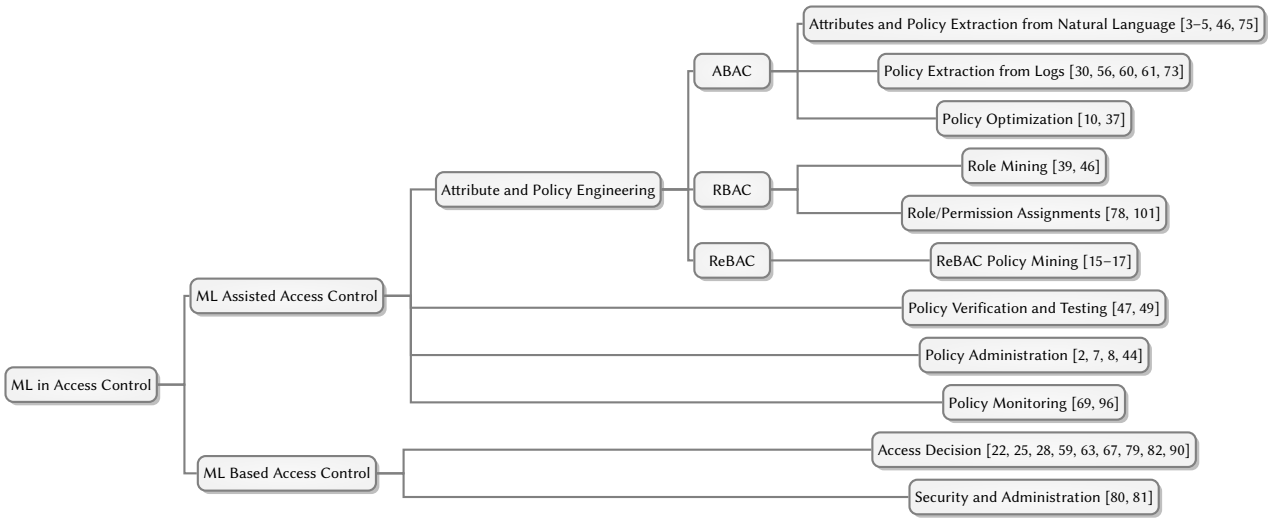


**Figure 2: A Taxonomy of Machine Learning in Access Control Domain.**

Narouei et al.[75] (2017) present a framework for ABAC that extracts security policies from natural language documents using deep recurrent neural networks (RNN). The model identifies access control policy content from high-level requirement documents, achieving a 5.58% improvement over other methods. However, human involvement is needed to accurately identify ACP sentences. Alohaly et al.[3] propose a deep learning framework using NLP, RE, and CNN to automate ABAC attribute extraction from NL policies. The framework identifies policy elements, extracts attribute values, and determines their categories, achieving an F1 score of 0.96 for subject attributes and 0.91 for object attributes. The evaluation uses datasets like iTrust[71], IBM Course Management App [1], CyberChair [95], and Collected ACP [97]. However, it lacks support for hierarchical ABAC systems. Alohaly et al.[5] enhance their previous framework[3] to automatically extract attributes from NL hierarchical ABAC policies using NLP and ML techniques. The multi-phase framework achieves an average F1 score of 0.96 for subject attributes and 0.91 for object attributes, evaluated on datasets like iTrust [71], IBM Course Management App [1], CyberChair [95], and Collected ACP [97].

Alohaly et al.[4] propose an automated process for extracting constraints in ABAC policies using NLP tools. The method uses BiLSTM models to identify and label conflicting factors in policy sentences, achieving an F1 score of 0.91 and detecting at least 75% of each constraint expression. The evaluation dataset includes 801 constraints in 747 NLACP sentences from various departments [4, 12, 13, 51]. Heaps et al.[46] developed a transformers-based deep learning model to extract access control information from user stories, including access control classification, named entity recognition, and access type classification[36]. Evaluated on the Dalpiaz dataset [32, 33], the model outperformed CNN and SVM, though CNN performed comparably in named entity recognition. The authors recommend a larger dataset for further improvement.

**Policy Extraction from Logs.** Mining ABAC policies from legacy systems is inefficient and laborious, making alternative sources beneficial. Additionally, maintaining these rules is challenging. A straightforward approach is to mine ABAC rules from access logs [98], as they reflect the existing access control policy.

Mocanu et al.[73] propose a deep learning approach using Restricted Boltzmann Machines (RBMs) to infer policies from logs, supporting negative authorization. The two-phase method generates candidate rules from logs and transforms them for comparison with Xu-Stoller[98]. Evaluated on a healthcare dataset, the approach shows promise but requires further implementation and evaluation for diverse real-world policies. Cotrini et al.[30] propose Rhapsody, an approach for mining ABAC rules from sparse access logs, addressing issues like rule size and over-permissiveness. Rhapsody modifies APRIORI-SD[62] to generate concise rules, evaluated on Amazon access logs [58, 93] and ETH Zurich lab logs. Using *universal cross-validation*, Rhapsody achieves higher F1 scores and better generalization compared to Classification Tree [14], CN2 [29], and other ABAC mining algorithms [62, 98].

Jabal et al.[56] introduce Polisma, a framework for learning ABAC policies using data mining, statistical, and ML techniques. Polisma generates, generalizes, and augments rules with restriction rules, then applies Random Forest (RF) and KNN classifiers to handle uncovered requests. Evaluated on real-world[93] and synthetic datasets, Polisma effectively develops accurate ABAC policies. Karimi et al.[60] propose an ABAC policy extraction method using access logs, building on their previous unsupervised learning approach[61]. The method uses K-modes clustering [21] to generate rules with positive and negative filters, followed by rule pruning and refinement. Evaluated on real-world and synthetic datasets, the approach effectively handles incomplete logs and noise but requires careful tuning of parameters for optimal performance.

**Policy Optimization.** Benkaouz et al. [10] propose using KNN algorithms for clustering and classifying ABAC policies, enhancing flexibility and reducing dimensionality in high-scale systems. The granularity of ABAC policies is adjusted by the parameter $k$, with smaller values for fine-grained models and larger values for coarse-grained models. This approach is still under development, with open

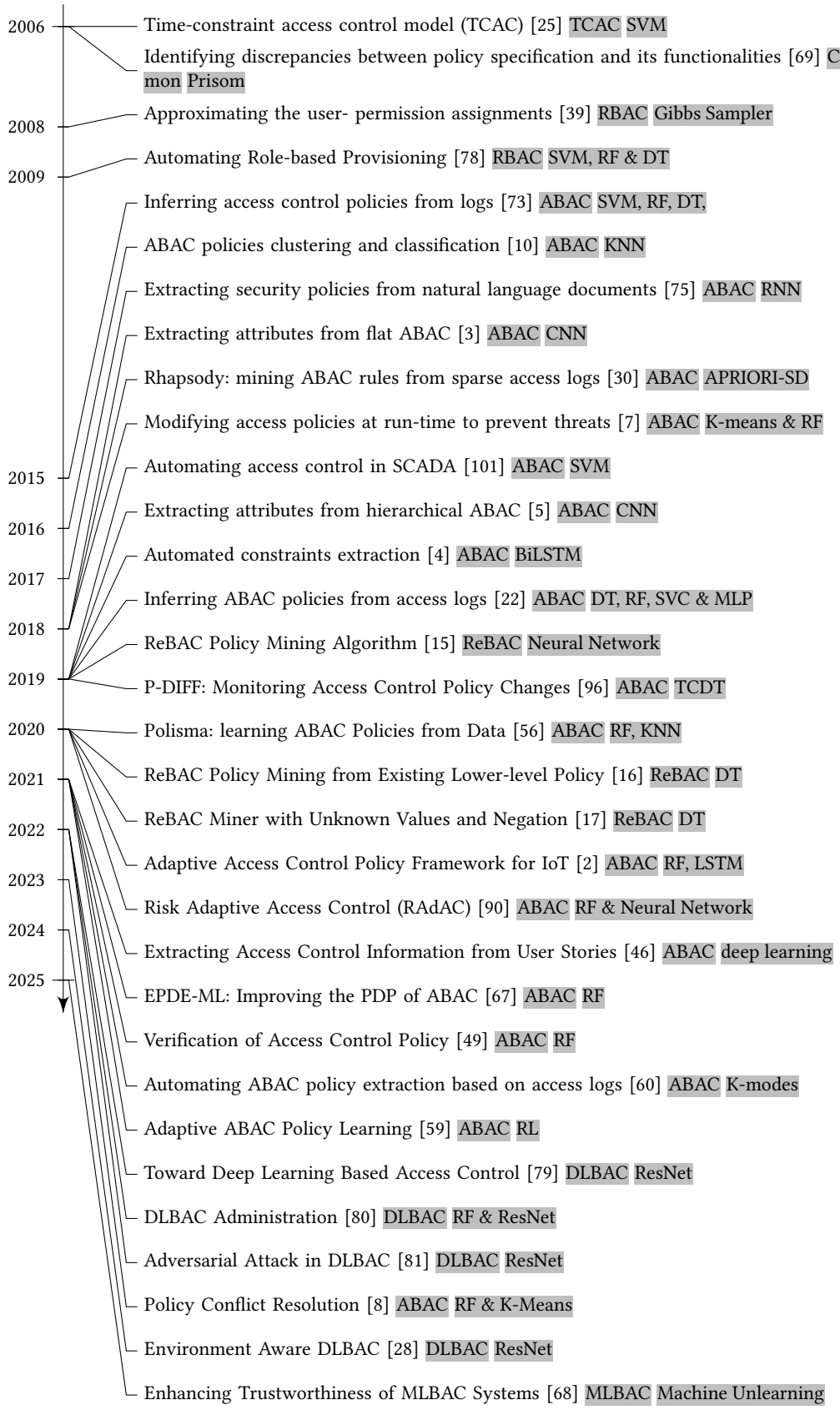


**Figure 3: A Timeline of Seminal Works Towards ML in Access Control. In each work, the first grey highlight indicates the access control model ('Common' implies the method is applicable for any access control model), and the second highlight denotes ML algorithms applied in the corresponding method.**

questions about the default value of *k*, the best KNN algorithms for clustering, and its applicability to various applications.

El Hadj et al.[37] propose ABAC-PC, a method for clustering ABAC policy rules based on decision effects and similarity scores, producing minimal representative rules. Extending Benkaouz et al.[10], the approach can reduce policy rules by up to 10% for policies with over 9000 rules and can be integrated with other tools to detect and resolve anomalies in XACML policies.

#### *3.1.2 Role Based Access Control (RBAC).*

**Role Mining.** Frank et al. [39] focus on bottom-up RBAC role mining to approximate user-permission assignments by finding minimal sets of roles, user-role, and role-permission assignments. They propose a probabilistic framework to address errors and non-meaningful roles in combinatorial algorithms, generalizing observations from existing user-permission assignments.

In another work, the authors use Gibbs sampler [77] for their Disjoint Decomposition Model, evaluated on synthetic and real-world datasets. The synthetic dataset includes 200 users, 200 permissions, ten business roles, and five technical roles, while the real-world dataset has 5000 users and 1323 permissions. The approach creates meaningful roles and identifies erroneous user-permission assignments, though it uniformly introduces errors in the synthetic dataset, unlike the unknown error count in real-world scenarios.

**Role/Permission Assignments.** Role-based provisioning is a standard in Identity Management products, but struggles in dynamic enterprises where frequent application reconfiguration and new service deployment are common. Adjusting role mappings to new privileges is challenging and costly, highlighting the need to reduce role maintenance efforts.

Ni et al. [78] propose a machine learning-based automated role maintenance system to provision existing roles with entitlements from new applications and new users with existing roles. The technique involves four phases: collecting role-entitlement mappings, filtering essential attributes, pre-processing data, and training classifiers. Evaluated on real-world and synthetic data, SVM was chosen as the final classifier, achieving FP rates between 0-5% and FN rates between 0-30%, with 70% of assignments automated and 30% needing assistance. Lu Zhou et al.[101] propose two ML-based approaches for automating role assignment in SCADA systems. They first apply SVM using static and dynamic attributes of users and devices for role assignments, but do not provide detailed evaluation results. They then experiment with the Adaboost algorithm using the same inputs, comparing real and discrete-valued Adaboost algorithms on a SCADA intrusion detection dataset[92].

#### *3.1.3 Relationship Based Access Control (ReBAC).*

Like ABAC policy mining approaches, ReBAC policy mining algorithms can also potentially reduce the effort to obtain a high-level policy from lower-level access control data.

Bui et al.[15] propose an efficient ReBAC policy mining algorithm, enhancing their previous evolutionary algorithm[20] with a neural network-based feature selection phase. This reduces search space and improves authorization mapping. Evaluated on real-world [34, 35] and synthetic policies, the enhanced algorithm is faster and more effective. Bui et al.[16] propose DTRM and DTRM$^-$, decision tree-based algorithms for mining ReBAC policies. DTRM mines ORAL policies, while DTRM$^-$ supports negative conditions and constraints. Evaluated on real-world[34, 35] and synthetic datasets [20], they produce smaller, faster policies than state-of-the-art methods [15, 55].

In most real-world data, information about permissions can be incomplete, or some attribute values can be missing (or unknown). Authors in [19, 30, 54, 65] solved different variants of the ABAC and ReBAC policy mining problem considering *incomplete permissions information*. However, all these works assume the attribute (and relationship in the case of ReBAC) information is complete (or known). The authors in [17] introduced DTRMU$^-$ and DTRMU algorithms for mining ABAC and ReBAC policies from ACLs, addressing incomplete information using Kleene's three-valued logic. This approach assigns a third truth value, 'U,' to unknown conditions, alongside true (T) and false (F) values. They developed a multi-way decision tree algorithm to classify authorization requests and generate ReBAC rules from labeled feature vectors [15, 16]. Experiments were conducted with sample policies from Bui et al.[20] and case studies from Decat et al.[34, 35], comparing the performance of mined rules to simplified original rules.

## 3.2 Policy Verification and Testing

Traditional policy verification methods are error-prone and time-consuming, lacking specificity in roles or permissions. They struggle with code-policy relations and new mappings. XACML-based policies also require rigorous verification to ensure accuracy and compliance.

Martin et al. [69] showed that ML algorithms can summarize policy properties and identify bug-exposing requests, revealing discrepancies between policy specifications and functionalities. Access requests are generated and applied to the system, with observations structured as request-response pairs. These pairs are used by an ML algorithm to infer policy properties and identify potential bugs. The authors integrated Sun's XACML implementation [83] and Weka [52] into their tool, which handles request generation, evaluation, and policy property inference. Using the Prism classification algorithm [23], the tool was tested on a university's grades repository policy [38]. Results showed that inferred properties effectively summarize the policy and identify bug-exposing requests.

Heaps et al.[47] propose leveraging deep learning to develop a more robust and efficient system. They suggest training a deep learning model based on links between code and policy elements. Since code elements lack numerical meaning, they use the Skip-gram Word2Vec algorithm[72] to embed code elements into a high-dimensional space. Experiments with JDK8 and Apache Shiro showed that this technique produces high-quality word embeddings and delivers state-of-the-art performance.

Access control policies are verified using model proof, data structure, system simulation, and test oracles. Comprehensive test case generation is challenging, so the NIST report [49] proposed a machine learning technique for efficient verification. This method trains a model based on policy rule attributes to generate a classification model, predicting access permissions and detecting inconsistencies. The authors used random forest (RF) as the ML method, encoding policy rules in a data table where each column represents an attribute, action, or permission, and each row represents a policy

**Table 1: Publicly Available Real-world Datasets Used in Access Control Researches. The names presented in the 'Name' column are chosen randomly.**

| SL. | Name | Publish Year | Reference | Type | Description | Application |
|---|---|---|---|---|---|---|
| 1.1 | IBM-CM | 2004 | IBM [1] | Access Policies | Natural language access control policy | [3], [5] |
| 1.2 | UniversityData | 2005 | Fisler et al. [38] | Access Policy | Central grades repository system for a university | [69] |
| 1.3 | Wikipedia | 2009 | Urdaneta et al. [94] | Access Logs | Access request traces from Wikipedia | [96] |
| 1.4 | AmazonUCI | 2011 | UCI Repository [93] | Access Logs | Access data of Amazon employees | [22], [30], [56], [60], [79] |
| 1.5 | iTrust | 2012 | Meneely et al. [71] | Access Policies | Natural language access control policy | [3], [5] |
| 1.6 | CyberChair | 2012 | Stadt et al. [95] | Access Policies | Natural language access control policy | [3], [5] |
| 1.7 | CollectedACP | 2012 | Xiao et al. [97] | Access Policies | Natural language access control policy collected from multiple sources | [3], [5] |
| 1.8 | AmazonKaggle | 2013 | Kaggle [58] | Access Logs | Two years historical access data of Amazon employees (12000 users and 7000 resources) | [22], [30], [59], [60], [67], [79], [8], [68] |
| 1.9 | eDocument | 2014 | Decat et al. [34] | Access Policy | e-document case study | [16], [17], [15] |
| 1.10 | Workforce | 2014 | Decat et al. [35] | Access Policy | Workforce management case study | [16], [17], [15] |
| 1.11 | SCADA-Intrusion | 2015 | Turnipseed et al. [92] | SCADA Data | SCADA dataset for intrusion detection system | [101] |
| 1.12 | Dalpiaz-UserStories | 2018 | Dalpiaz et al. [32, 33] | User Stories | Over 1600 user stories from 21 web applications | [46] |
| 1.13 | Incident | 2018 | Amaral et al. [6] | Event Logs | Event log from an incident management process | [22] |

rule. Rules with multiple actions or object attributes are split into sub-rules. The RF model is evaluated to detect permission conflicts and ensure it recognizes policy rule semantics, including condition, separation of duty, and exclusion properties. The accuracy function analysis indicates the semantic correctness of the policy. Less than 100% correctness suggests conflict rules may exist. Overall, the algorithm efficiently verifies policies and detects inconsistencies.

Table 3 summarizes access control policy verification tools and methods using machine learning.

## 3.3 Policy Administration

Adapting access control policies to tackle cyber attacks is challenging due to their static nature. Regular maintenance is required to keep policies up-to-date, which, if not automated, is laborious and error-prone [7]. Erroneous policies can make the system vulnerable to adversaries or misuse by internal users. Therefore, reinforcing the access control system to identify misconfigurations and adjust policies accordingly is crucial.

Manual access control policy updates are laborious and error-prone [7], making systems vulnerable to cyber attacks. Authors in [7] present ML-AC, an ML-based approach that updates policies automatically at run-time to prevent such threats. ML-AC monitors user access behavioral features (e.g., frequency, data amount, location) and adjusts access control rules based on contextual knowledge. This method includes a Contextual Behaviour Learning component in the Policy Administration Point (PAP) to build user profiles and adjust policies. ML-AC uses RF to classify access control behavior as normal or anomalous and refines policies by encoding ML-rules with learned contextual knowledge. It also monitors user behavior evolution using Olindda [89] to detect new clusters. Experiments with a synthetic dataset demonstrated ML-AC's effectiveness. Comparisons with BBNAC [40] and ML-AC$_{nok}$ (without contextual knowledge) showed ML-AC's superior performance.

Alkhresheh et al. [2] proposed an adaptive access control policy framework for IoT, refining policies based on device behaviors. They suggest a policy management module for adaptation, including behavior classification and policy refinement, alongside the traditional ABAC server. Both servers include a context monitor. Using RF and LSTM [48] on three years of access data from a university's door locking system, they found LSTM outperformed RF with larger datasets due to its ability to learn from longer sequences. The study concluded that LSTM scales better in IoT environments.

Gumma et al. [44] proposed PAMMELA, an ML-based ABAC policy administration method that creates new rules and extends existing policies. It operates in two phases: training an ML classifier on ABAC policy rules and generating rules based on access requests. The authors experimented with three datasets containing various

**Table 2: Summarizing Machine Learning Based Policy Mining. The dataset type 'RW', 'RWA', and 'Syn' indicates Real-World, Real-World Augmented, and Synthetic dataset, respectively. We link Dataset Type column to 'SL.' column of Table 1 if the respective dataset is public. We follow the same convention in Table 3, 4, and 5.**

| Reference | Application | Problem Considered | Access Control Model | ML Approach | Dataset Type |
|---|---|---|---|---|---|
| Frank et al. 2008 [39] | Not specified | Probabilistic bottom-up approaches for RBAC role mining | RBAC | Gibbs sampler & Disjoint Decomposition | Syn & RW |
| Ni et al. 2009 [78] | Not specified | Adjusts roles and permission mapping | RBAC | SVM (and others including DT, RF) | Syn & RW |
| Mocanu et al. 2015 [73] | Healthcare | Policy inference from logs | ABAC | Restricted Boltzmann Machines | Syn |
| Benkaouz et al. 2016 [10] | Not specified | Classification and clustering of policies | ABAC | K-Nearest Neighbors | Not Used |
| Narouei et al. 2017 [75] | Not specified | Policy extraction from natural language documents | ABAC | Recurrent Neural Network | Syn |
| El Hadj et al. 2017 [37] | Not specified | Classification and clustering of policies | ABAC | K-Nearest Neighbors | Syn |
| Alohaly et al. 2018 [3] | Not specified | ABAC attribute extraction from natural language | ABAC | CNN | RWA (SL: 1.1, 1.5, 1.6, 1.7) |
| Karimi et al. 2018 [61] | Healthcare, education | Policy extraction | ABAC | K-modes | Syn |
| Cotrini et al. 2018 [30] | Not specified | Policy mining | ABAC | APRIORI-SD [62] | Syn & RW (SL: 1.4, 1.8) |
| Alohaly et al. 2019 [5] | Not specified | Attribute extraction from natural language for ABAC | Hierarchical ABAC | CNN | RWA (SL: 1.1, 1.5, 1.6, 1.7) |
| Alohaly et al. 2019 [4] | Not specified | ABAC constraints extraction from natural language policies | ABAC | BiLSTM | RWA |
| Zhou et al. 2019 [101] | SCADA | Role and permission assignments | RBAC | SVM & Adaboost | RWA (SL: 1.11) |
| Bui et al. 2019 [15] | Not specified | Policy mining | ReBAC | Neural Network | Syn & RW (SL: 1.9, 1.10) |
| Bui et al. 2020 [16] | Not specified | Policy mining | ReBAC | Decision Tree | Syn & RW (SL: 1.9, 1.10) |
| Bui et al. 2020 [17] | Not specified | Policy mining | ABAC, ReBAC | Decision Tree | Syn & RW (SL: 1.9, 1.10) |
| Jabal et al. 2020 [56] | Not specified | Learns ABAC policies from logs | ABAC | RF, KNN | Syn & RW (SL: 1.4) |
| Karimi et al. 2021 [60] | Not specified | Automating ABAC policy extraction based on access logs | ABAC | K-modes | Syn & RW (SL: 1.4, 1.8) |
| Heaps et al. 2021 [46] | Not specified | Extracting access control policy from user stories | RBAC and ABAC | Transformers, CNN, SVM | RW (SL: 1.12) |

**Table 3: Summarizing Machine Learning Based Policy Verification. 'Common': any access control model.**

| Reference | Application | Problem Considered | Access Control Model | ML Approach | Dataset Type |
|---|---|---|---|---|---|
| Martin et al. 2006 [69] | Not Specified | Inferring policy and identifying bug-exposing requests | Common | Prism [23] | RW (SL: 1.2) |
| Heaps et al. 2019 [47] | Not Specified | Policy verification automation | RBAC | Neural Network | Syn |
| Vincent C. Hu 2021 [49] | Not Specified | Verifying access control policy | Common | RF | Syn |

ABAC rules, subject and object attributes, and access requests. They evaluated PAMMELA using neural networks, decision trees, RF, Extra Trees (ET)[42], Gradient Boosting (GB)[41], and Extreme Gradient Boosting (XGB) [26]. The study provided insights into managing ABAC policies using PAMMELA.

Recently, Ayedh et al. [8] introduced an enhanced and distributed access control decision-making model that utilizes a random forest

**Table 4: Summarizing Machine Learning Based Policy Administration, Monitoring, and Auditing. 'Common': any access control model.**

| Reference | Application | Problem Considered | Access Control Model | ML Approach | Dataset Type |
|---|---|---|---|---|---|
| Argento et al. 2018 [7] | Not Specified | Improving ABAC policy administration point (PAP) | ABAC & Common | RF and K-means Clustering | Syn |
| Xiang et al. 2019 [96] | Not Specified | Continuous access control validation and forensics | Common | Time-Changing Decision Tree (TCDT) | RW (SL: 1.3) |
| Ashraf et al. 2020 [2] | IoT | Refining access policies | ABAC | RF and RNN | RW |
| Gumma et al. 2021 [44] | Not Specified | ABAC policy administration | ABAC | Neural Network, DT, RF, etc. | Syn |
| Ayedh et al. 2023 [8] | BYOD environments | Policy conflict resolution | ABAC | RF and K-Means | RW (SL: 1.8) |

algorithm to effectively resolve policy conflicts in BYOD environments. By dynamically adapting to diverse device profiles and network conditions, the proposed model achieves accurate and efficient access decisions while minimizing administrative overhead.

Table 4 outlines related methods proposed for the access control administration using machine learning.

### 3.4 Policy Monitoring and Auditing

Regular access control policy updates are challenging and error-prone [96], risking severe security incidents. Xiang et al. [96] developed P-DIFF to help system admins monitor policy changes and investigate malicious access by backtracking related changes. They proposed a Time-Changing Decision Tree (TCDT) to handle time-series information, modeling access control behavior over time. P-DIFF uses access logs to generate a TCDT, aiding policy change validation and forensics analysis. Experiments with datasets from five real-world systems, including Wikipedia [94], showed P-DIFF detects 86 to 100% of policy changes with 89% precision and 85 to 98% efficacy in forensic analysis. Table 4 further outlines this work.

## 4 ML-BASED ACCESS CONTROL (MLBAC)

Recent research highlights the benefits of using ML models for accurate access control decision-making [22, 25, 59, 67, 79, 90]. These systems use trained ML models instead of language-based policies to decide access (grant or deny) based on user and resource metadata and attributes. Metadata and attributes are features that the ML model learns for subsequent access decisions. Besides, it is required to administer those trained ML models such that any changes in access policy can be accommodated [80], while securing them from external attack [81]. We briefly discuss these approaches below and summarize them in Table 5.

### 4.1 Access Decision

Access control policies sometimes need to be restricted by access hours. For example, a user may be denied access outside office hours. Chang et al.[25] proposed a time-constraint access control system using SVM, divided into three phases: input pattern transforming, training SVMs, and authority decision. They implemented SVMs using LIBSVM[24], training on users' login times and passwords to classify users and grant access based on these factors. The system uses trained SVMs for access decisions instead of traditional policies. Performance evaluation with training data showed the system can authenticate users' access rights, demonstrating its practicality.

Centralized access control architectures with static policies are limited in IoT due to devices' computational constraints. Outsourcing access control management introduces security and privacy concerns. IoT needs a framework suitable for its distributed nature, allowing user privacy control and centralized handling. Outchakoucht et al. [82] proposed a blockchain and ML-based access control approach for IoT. The approach uses blockchain for distributed policy management and ML, specifically Reinforcement Learning, to dynamically adjust access control policies as resources are accessed and security policies executed.

Cappelletti et al.[22] explored symbolic (DT, RF) and non-symbolic (SVC, MLP) ML techniques for inferring ABAC policies from access logs. They used two Amazon datasets[58, 93] and the Incident dataset [6], noting the sparsity and imbalance of Amazon datasets versus the balanced Incident dataset. PCA and t-SNE [99] revealed well-separated clusters in the Incident log but not in the Amazon dataset. Amazon datasets showed around 50% accuracy due to sparsity, while Incident logs had high accuracy. MLP excelled with sparse data, capturing complex relationships. Symbolic techniques offered better decision understanding, highlighting concerns about black-box ML techniques' explainability and verifiability.

Khilar et al. [63] proposed a trust-based cloud resource access approach using user access history and behavior, considering bogus, unauthorized, and forbidden requests. They tested ML techniques like KNN, decision tree, logistic regression, naive Bayes, neural networks, and ensemble algorithms. The ensemble model of random forest (RF) and K-nearest neighbor performed best, with neural networks achieving the highest performance.

Srivastava et al. [90] proposed RAdAC, a framework assessing requester genuineness, calculating risk, and acting accordingly, using attributes like access time, location, request frequency, and resource sensitivity. They developed a Hospital Management System prototype and tested a neural network and RF algorithm. RF performed best with both input data and engineered parameters, highlighting the need for domain expertise to determine optimal values.

Liu et al. [67] proposed EPDE-ML, an Efficient Permission Decision Engine scheme based on ML to improve the ABAC model's policy decision point (PDP). EPDE-ML uses an RF algorithm trained on user attributes and prior access control information to permit or

**Table 5: Summarizing Machine Learning Based Access Control Decision and Administration**

| Reference | Application | Problem Considered | Access Control Model | ML Approach | Dataset Type |
|---|---|---|---|---|---|
| Chang et al. 2006 [25] | Not specified | A novel access control model with time constraint | Time-constraint Access Control | SVM | Syn |
| Outchakoucht et al. 2017 [82] | IoT | Blockchain based access control policy | Not Applicable | Reinforcement Learning | No Evaluation |
| Cappelletti et al. 2019 [22] | Not specified | Inferring ABAC policies from access logs | ABAC | DT, RF, SVM, MLP | RW (SL: 1.4, 1.8, 1.13) |
| Khilar et al. 2019 [63] | Cloud Computing | Policy for cloud resources | Trust-Based Access Control | RF, DT, SVM, Neural Network, etc. | Not Specified |
| Srivastava et al. 2020 [90] | Defense, airport, and healthcare | Novel access control framework | Risk Adaptive Access Control (RAdAC) | Neural Network, RF | Not Specified |
| Liu et al. 2021 [67] | Big Data & IoT | Improves the policy decision point (PDP) of the ABAC model | ABAC | RF | RW (SL: 1.8) |
| Karimi et al. 2021 [59] | IoT | Adaptive ABAC policy learning | ABAC | Reinforcement Learning | Syn & RW (SL: 1.8) |
| Nobi et al. 2022 [79] | Not specified | DLBAC | DLBAC | ResNet | Syn & RW (SL: 1.4, 1.8) |
| Nobi et al. 2022 [80] | Not specified | DLBAC administration | DLBAC | RF & ResNet | Syn |
| Nobi et al. 2022 [81] | Not specified | Adversarial attack in DLBAC | DLBAC | ResNet | Syn |
| Chhetri et al. 2024 [28] | IoT | Environment aware DLBAC | DLBAC | ResNet | Syn |
| Llamas et al. 2025 [68] | Not specified | To enhance the trustworthiness of MLBAC systems | MLBAC | Machine Unlearning | RW (1.8) |

deny access requests. The process is split into two phases: Phase 1 uses the trained model for decisions, and Phase 2 updates the model with current policy information. Experiments on Amazon's access control policy set [58] showed EPDE-ML's superior performance, with an AUC of 0.975 and 92.6% accuracy. Decision time remained consistent at around 0.115 seconds, regardless of policy size.

Karimi et al. [59] proposed an adaptive access control approach using RL to address challenges like limited labeled data and sparse logs. The goal is to develop an adaptive ABAC policy learning model for smart home IoT environments, with methods to speed up learning based on attribute value hierarchy. Experiments on synthesized and real datasets, including Amazon [58], showed the effectiveness of the approach. The real dataset included employee access requests, indicating whether access was permitted, along with attribute values and resource identifiers.

Traditional access control systems struggle in dynamic, large-scale environments, making it difficult for human administrators to maintain accurate access control states. Nobi et al.[79] propose Deep Learning Based Access Control (DLBAC), which uses user and resource metadata directly, eliminating the need for attribute and policy engineering. The DLBAC model, based on ResNet[45], outputs a trained neural network for access control decisions. The authors developed a DLBAC prototype, tested it with eight synthetic datasets and two Amazon datasets [58, 93], and found that DLBAC makes more accurate and generalized access control decisions than traditional systems. DLBAC better balances over-provision (unauthorized access) and under-provision (denied access) inefficiencies compared to policy-based systems. However, DLBAC may inherit biases from training data, potentially leading to adverse decisions. For example, in the Amazon dataset, where most authorization tuples involve grant decisions, DLBAC might favor similar decisions. Ensuring fair decisions requires auditing the training data and evaluating decisions for fairness.

Chhetri et al. [28] introduce an environment-aware access control model that leverages deep learning to integrate contextual environmental data in security decision-making. By analyzing real-time conditions and adapting policies accordingly, the model enhances precision in identifying legitimate access requests while strengthening overall system resilience in dynamic settings.

## 4.2 Administration and Security

Nobi et al. [80] investigate the administration challenges of ML-based access control systems, particularly in capturing changes in access control states. They compare the performance of symbolic (RF) and non-symbolic (ResNet) ML methods in a simulated environment. The study highlights the advantages and disadvantages of both approaches, such as insufficient learning of new changes and forgetting existing access information. Experimental results indicate that non-symbolic methods outperform symbolic ones in adapting to incremental changes in access control states. Recently, Llamas et al. [68] introduced a certified unlearning framework for ML-based access control, ensuring compliance with evolving policies by removing specific data without full retraining. This enhances adaptability, security, and privacy in access control systems.

While deploying an ML model for access decision, it is crucial to secure the model from unwanted intervene. Nobi et al. [81] examine the security vulnerabilities of ML-based access control systems, particularly their susceptibility to adversarial attacks. It highlights

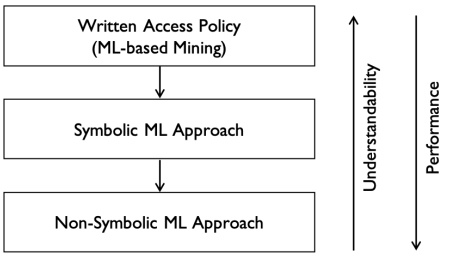


**Figure 4: Trade-off Between Performance and Explainability in ML-based Systems.**

how minor input modifications can lead to incorrect access decisions by ML models. The study focuses on manipulating user and resource information to gain unauthorized access, using ResNet models in simulated environments. Results show that adversarial attacks can be designed for these models, but access control-specific constraints can mitigate such attacks to some extent.

# 5 OPEN CHALLENGES AND FUTURE RESEARCH DIRECTIONS

## 5.1 Understanding Access Control Decisions

In complex situations where access control systems overlap, non-symbolic ML models often make better decisions than traditional policies or symbolic ML models [22]. Neural networks and other non-symbolic methods can learn subtle differences among users, resources, and their relationships. However, achieving superior performance with these models often comes at the cost of *explainability*—understanding the reasons behind specific access decisions.

While ML models for access control are still in their early stages, a lack of explainability could hinder their growth. Written policies and symbolic ML models offer straightforward explanations through human-understandable logical rules. For example, symbolic ML methods allow easy extraction of logical rules from decision trees. In contrast, non-symbolic ML models, such as DLBAC [79] or Karimi et al. [59], use 'black box' functions that are not easily interpretable, which is crucial for security-sensitive domains.

As shown in Figure 4, moving from written policies to non-symbolic approaches increases performance but decreases explainability. This limitation affects other domains, including computer vision, malware analysis, and financial systems. The issue of explainability is a very active research area in ML [53]. Solutions are often domain-specific; for example, a method for explaining computer vision models may not work for access control.

Nobi et al. [79] addressed this issue in access control by proposing two methods for explaining neural network-based decisions in human terms. However, these methods do not guarantee 100% accuracy in understanding decisions. Therefore, there is significant potential for further research to improve explanations and enhance intuition in this area.

## 5.2 Access Control Administration

Maintaining access control systems, whether traditional or ML-based, is crucial for long-term security. This involves modifying policy configurations or attributes to accommodate authorization changes. RBAC tasks include managing permissions, roles, and hierarchies [11, 84, 86]. ABAC involves adjusting attributes and rules [57, 85], while ReBAC requires managing relationships and policies [27, 91]. These issues are well-studied in traditional models.

ML-based access control introduces new challenges, such as updating ML models and access information. Nobi et al. [80] defined administration requirements for ML-based systems and proposed updating methods. Further investigation is needed to address remaining challenges.

## 5.3 Adversarial Attacks

Adversarial attacks are a common concern for ML-based systems. An adversary can obtain unwarranted decisions [9] by fooling the network with adversarial samples indistinguishable from natural ones [100]. In ML-based access control, adversarial attacks can force systems to grant unauthorized access. Attackers can *trick* the system by providing manipulated user and resource information. Additionally, *attribute-hiding attacks* may occur, where attackers hide or remove portions of their information to secure access. Nobi et al. [81] explored this issue, demonstrating that access to ML models creates potential for adversarial attacks. They showed that such attacks can be mitigated using access control-specific constraints, but their work is limited to scenarios where attackers cannot access the deployed ML model. No other methods addressing adversarial issues in access control were found.

Therefore, it is crucial to investigate adversarial attacks more thoroughly from an access control perspective and develop solutions to protect systems against these vulnerabilities.

## 5.4 Lack of 'Good' Datasets

Ground truth information is crucial for evaluating access control applications like role mining and policy mining, especially for ML-based methods [74]. However, few high-quality datasets are available, and existing ones are often imbalanced [30, 79] and lack sufficient information [46]. This makes designing effective systems challenging. Researchers address this by using data preprocessing [67], data augmentation [3, 4, 101], and synthetic datasets [98]. A hybrid approach combining real-world and synthetic datasets is also common [17, 56, 60, 79].

Despite the rise in ML applications for access control, the lack of quality datasets from real-world organizations remains a significant obstacle. Many datasets are anonymous or incomplete and lack the necessary semantics and granularity [74]. High-quality datasets are essential for advancing ML-based access control.

## 5.5 Bias and Fairness

In access control, an ML model is over-provisioned if it is biased toward granting unauthorized access. Conversely, it is under-provisioned if it denies desired access. Both can be measured quantitatively as high FPR and low TPR, respectively [79]. These biases often arise from imbalanced training data or improper ML model design.

For example, the Amazon dataset [58] contains two years of historical access data, where employees were manually allowed or denied access to resources. The data is highly imbalanced, with over 90% of requests granted access [30, 79]. This disproportionate training data biases the model towards granting access [79]. Bias can also result from poor ML model/algorithm design. Therefore, understanding the characteristics of training data and ML algorithms is

crucial for developing a fair and reliable system [70]. Additionally, besides evaluating access control decisions, it is important to assess fairness performance and establish a feedback loop [31, 79].

### 5.6 Insufficient Tools for Verification

A reliable system design is ensured through rigorous *testing* and *verification*. Testing evaluates a system under various conditions to observe behavior and detect errors, while verification ensures the system does not misbehave under general circumstances [43]. In access control, policies are verified and tested similarly to software functionality [50]. Ensuring correct access control decisions is complex and requires significant effort. Failure to verify the system's correctness can lead to serious consequences, such as over-provision, under-provision, and adversarial attacks. This area is well-studied for traditional access control systems, with established verification methods [49, 64].

When ML is applied, performance is measured using *unseen data* to test model correctness. However, this method cannot identify all possible misclassifications, and the impact of misclassification varies across domains. For example, granting unauthorized access can be more costly than denying legitimate access in access control. Therefore, comprehensive verification is crucial before deploying ML-assisted access control systems. While there are methods to verify ML models automatically [66], each has its pros and cons. Further research is needed to design a systematic verification and testing framework for ML and access control.

## 6 SUMMARY

This work comprehensively explores the exciting intersection of access control and machine learning. We propose a taxonomy of machine learning for access control and discuss each approach within this framework. Our findings reveal that machine learning is making significant strides in various areas of access control, including attribute engineering, policy mining, and access control policy verification. We also examine efforts to use trained machine learning models to make access control decisions, replacing traditional written policies. These models offer robust and generalized decision-making, though they often lack transparency in how decisions are made. Additionally, we outline publicly available real-world datasets used in machine learning-based access control research.

Finally, we share our observations and vision regarding open challenges in the domain, providing potential guidelines to overcome them. This work aims to shed light on the promising future of machine learning in access control and inspire further research and innovation.

## REFERENCES


[1] IBM (2004). 2004. Course registration requirements. Retrieved August, 2021 from https://khanhn.files.wordpress.com/2016/08/vidu-ibm.pdf

[2] Ashraf Alkhresheh et al. 2020. Adaptive access control policies for IoT deployments. In *IEEE IWCMC*.

[3] Manar Alohaly et al. 2018. A deep learning approach for extracting attributes of ABAC policies. In *ACM SACMAT*.

[4] Manar Alohaly et al. 2019. Towards an Automated Extraction of ABAC Constraints from Natural Language Policies. In *IFIP International Conference on ICT Systems Security and Privacy Protection*. Springer.

[5] Manar Alohaly, Hassan Takabi, and Eduardo Blanco. 2019. Automated extraction of attributes from natural language attribute-based access control (ABAC) policies. *Cybersecurity* (2019).

[6] C. AL Amaral, Marcelo Fantinato, Hajo A Reijers, and Sarajane M Peres. 2018. Enhancing completion time prediction through attribute selection. In *Information Technology for Management: Emerging Research and Applications*. Springer.

[7] Luciano Argento et al. 2018. Towards adaptive access control. In *DBSec*. Springer.

[8] Aljuaid Turkea Ayedh M, Ainuddin Wahid Abdul Wahab, and Mohd Yamani Idna Idris. 2023. Enhanced adaptable and distributed access control decision making model based on machine learning for policy conflict resolution in BYOD environment. *Applied Sciences* 13, 12 (2023), 7102.

[9] Vincent Ballet et al. 2019. Imperceptible adversarial attacks on tabular data. *arXiv* (2019).

[10] Yahya Benkaouz, Mohammed Erradi, and Bernd Freisleben. 2016. Work in progress: K-nearest neighbors techniques for abac policies clustering. In *ACM International Workshop on Attribute Based Access Control*.

[11] Rafae Bhatti, Basit Shafiq, Elisa Bertino, Arif Ghafoor, and James BD Joshi. 2005. X-gtrbac admin: A decentralized administration model for enterprise-wide access control. *ACM TISSEC* (2005).

[12] Khalid Zaman Bijon et al. 2013. Constraints specication in attribute based access control. *Science* (2013).

[13] Khalid Zaman Bijon, Ram Krishnan, and Ravi Sandhu. 2013. Towards an attribute based constraints specification language. In *IEEE International Conference on Social Computing*.

[14] Leo Breiman et al. 2017. *Classification and regression trees*. Routledge.

[15] Thang Bui et al. 2019. Efficient and extensible policy mining for relationship-based access control. In *ACM SACMAT*.

[16] Thang Bui and Scott D Stoller. 2020. A decision tree learning approach for mining relationship-based access control policies. In *ACM SACMAT*.

[17] Thang Bui and Scott D Stoller. 2020. Learning Attribute-Based and Relationship-Based Access Control Policies with Unknown Values. In *International Conference on Information Systems Security*. Springer.

[18] Thang Bui, Scott D Stoller, and Jiajie Li. 2017. Mining relationship-based access control policies. In *ACM SACMAT*.

[19] Thang Bui, Scott D Stoller, and Jiajie Li. 2018. Mining relationship-based access control policies from incomplete and noisy data. In *International Symposium on Foundations and Practice of Security*. Springer.

[20] Thang Bui, Scott D Stoller, and Jiajie Li. 2019. Greedy and evolutionary algorithms for mining relationship-based access control policies. *Computers & Security* (2019).

[21] F. Cao et al. 2009. A new initialization method for categorical data clustering. *Expert Systems with Applications*.

[22] Luca Cappelletti, Stefano Valtolina, Giorgio Valentini, Marco Mesiti, and Elisa Bertino. 2019. On the quality of classification models for inferring ABAC policies from access logs. In *IEEE International Conference on Big Data*. IEEE.

[23] Jadzia Cendrowska. 1987. PRISM: An algorithm for inducing modular rules. *International Journal of Man-Machine Studies* (1987).

[24] Chih-Chung Chang et al. 2011. LIBSVM: a library for support vector machines. *ACM TIST* (2011).

[25] Chin-Chen et al. Chang. 2006. An Access Control System with Time-constraint Using Support Vector Machines. *Int. J. Netw. Secur.* (2006).

[26] Tianqi Chen et al. 2015. Xgboost: extreme gradient boosting. *R package version 0.4-2* (2015).

[27] Yuan Cheng, Khalid Bijon, and Ravi Sandhu. 2016. Extended ReBAC administrative models with cascading revocation and provenance support. In *ACM SACMAT*.

[28] Pankaj Chhetri, Smriti Bhatt, Paras Bhatt, Mohammad Nur Nobi, James Benson, and Ram Krishnan. 2024. Environment Aware Deep Learning Based Access Control Model. In *Proceedings of the 2024 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems*. 81–89.

[29] Peter Clark and Tim Niblett. 1989. The CN2 induction algorithm. *Machine learning* (1989).

[30] Carlos Cotrini et al. 2018. Mining ABAC rules from sparse logs. In *IEEE EuroS&P*.

[31] Christine M Cutillo et al. 2020. Machine intelligence in healthcare—perspectives on trustworthiness, explainability, usability, and transparency. *NPJ digital medicine* (2020).

[32] F Dalpiaz. 2018. Requirements data sets (user stories). *Mendeley Data, v1* (2018). https://doi.org/10.17632/7zbk8zsd8y.1

[33] Fabiano Dalpiaz, Ivor Van der Schalk, and Garm Lucassen. 2018. Pinpointing ambiguity and incompleteness in requirements engineering via information visualization and NLP. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer.

[34] Maarten Decat, Jasper Bogaerts, Bert Lagaisse, and Wouter Joosen. 2014. The e-document case study: functional analysis and access control requirements. *CW Reports CW654, Department of Computer Science, KU Leuven* (2014).

[35] Maarten Decat, Jasper Bogaerts, Bert Lagaisse, and Wouter Joosen. 2014. The workforce management case study: functional analysis and access control requirements. *CW Reports CW655, Dept. of Computer Science, KU Leuven* (2014).

[36] Jacob Devlin et al. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*.

[37] Maryem Ait El Hadj et al. 2017. ABAC rule reduction via similarity computation. In *ICNS*. Springer.

[38] Kathi Fisler et al. 2005. Verification and change-impact analysis of access-control policies. In *ICSE*.

[39] Mario Frank et al. 2008. A class of probabilistic models for role engineering. In *ACM CCS*.

[40] Vanessa Frias-Martinez et al. 2009. A network access control mechanism based on behavior profiles. In *IEEE ACSAC*.

[41] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001).

[42] Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine learning* (2006).

[43] Ian Goodfellow et al. 2017. The challenge of verification and testing of machine learning. *Cleverhans-blog* (2017).

[44] Varun Gumma et al. 2021. PAMMELA: Policy Administration Methodology using Machine Learning. *arXiv*.

[45] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[46] John Heaps, Ram Krishnan, Yufei Huang, Jianwei Niu, and Ravi Sandhu. 2021. Access Control Policy Generation from User Stories Using Machine Learning. In *DBSec*. Springer.

[47] John Heaps, Xiaoyin Wang, Travis Breaux, and Jianwei Niu. 2019. Toward Detection of Access Control Models from Source Code via Word Embedding. In *ACM SACMAT*.

[48] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* (1997).

[49] Vincent Hu. 2021. *Machine Learning for Access Control Policy Verification*. Technical Report. NIST.

[50] Vincent C Hu and Rick Kuhn. 2016. Access control policy verification. *Computer* (2016).

[51] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* (2015).

[52] H Witten Ian and Frank Eibe. 2005. Data Mining: Practical machine learning tools and techniques.

[53] Mir Riyanul Islam, Mobyen Uddin Ahmed, Shaibal Barua, and Shahina Begum. 2022. A systematic review of explainable artificial intelligence in terms of different application domains and tasks. *Applied Sciences* (2022).

[54] Padmavathi Iyer et al. 2020. Active learning of relationship-based access control policies. In *ACM SACMAT*.

[55] Padmavathi Iyer and Amirreza Masoumzadeh. 2019. Generalized mining of relationship-based access control policies in evolving systems. In *ACM SACMAT*.

[56] Amani Abu Jabal, Elisa Bertino, Jorge Lobo, Mark Law, Alessandra Russo, Seraphin Calo, and Dinesh Verma. 2020. Polisma-a framework for learning attribute-based access control policies. In *ESORICS*. Springer.

[57] Sadhana Jha, Shamik Sural, Vijayalakshmi Atluri, and Jaideep Vaidya. 2016. An administrative model for collaborative management of ABAC systems and its security analysis. In *IEEE Conference on Collaboration and Internet Computing*.

[58] Kaggle. 2013. Amazon Employee Access Challenge. https://www.kaggle.com/c/amazon-employee-access-challenge/

[59] Leila Karimi et al. 2021. Adaptive ABAC Policy Learning: A Reinforcement Learning Approach. *arXiv* (2021).

[60] Leila Karimi et al. 2021. An automatic attribute based access control policy extraction from access logs. *IEEE TDSC*.

[61] Leila Karimi and James Joshi. 2018. An unsupervised learning based approach for mining attribute based access control policies. In *2018 IEEE International Conference on Big Data (Big Data)*. IEEE.

[62] Branko Kavšek and Nada Lavrač. 2006. APRIORI-SD: Adapting association rule learning to subgroup discovery. *Applied Artificial Intelligence* (2006).

[63] Pabitr Mohan Khilar, Vijay Chaudhari, and Rakesh Ranjan Swain. 2019. Trust-based access control in cloud computing using machine learning. In *Cloud Computing for Geospatial Big Data Analytics*. Springer.

[64] D Richard Kuhn et al. 2016. Pseudo-exhaustive testing of attribute based access control rules. In *IEEE ICSTW*.

[65] Mark Law, Alessandra Russo, Elisa Bertino, Krysia Broda, and Jorge Lobo. 2020. Fastlas: scalable inductive logic programming incorporating domain-specific optimisation criteria. In *AAAI Conference on AI*.

[66] F. Leofante et al. 2018. Automated verification of neural networks: Advances, challenges and perspectives. *arXiv*.

[67] Aodi Liu, Xuehui Du, and Na Wang. 2021. Efficient Access Control Permission Decision Engine Based on Machine Learning. *Security and Communication Networks* (2021).

[68] Javier Martínez Llamas, Davy Preuveneers, and Wouter Joosen. 2025. Certified unlearning for a trustworthy machine learning-based access control administration. *International Journal of Information Security* 24, 2 (March 2025), 94. https://doi.org/10.1007/s10207-025-01003-5

[69] Evan Martin and Tao Xie. 2006. Inferring access-control policy properties via machine learning. In *IEEE POLICY*.

[70] Ninareh Mehrabi et al. 2019. A survey on bias and fairness in machine learning. *arXiv* (2019).

[71] Andrew Meneely, Ben Smith, and Laurie Williams. 2012. Appendix B: iTrust electronic health care system case study. *Software and Systems Traceability* (2012).

[72] Tomas Mikolov et al. 2013. Efficient estimation of word representations in vector space. *arXiv* (2013).

[73] Decebal Mocanu, Fatih Turkmen, Antonio Liotta, et al. 2015. Towards ABAC policy mining from logs with deep learning. In *Proceedings of the 18th International Multiconference, ser. Intelligent Systems*.

[74] Ian Molloy, Jorge Lobo, and Suresh Chari. 2011. Adversaries' holy grail: access control analytics. In *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*.

[75] Masoud Narouei, Hamed Khanpour, Hassan Takabi, Natalie Parde, and Rodney Nielsen. 2017. Towards a top-down policy engineering framework for attribute-based access control. In *ACM SACMAT*.

[76] Masoud Narouei and Hassan Takabi. 2015. Towards an automatic top-down role engineering approach using natural language processing techniques. In *ACM SACMAT*.

[77] Radford M Neal. 2000. Markov chain sampling methods for Dirichlet process mixture models. *Journal of computational and graphical statistics* (2000).

[78] Qun Ni et al. 2009. Automating role-based provisioning by learning from examples. In *ACM SACMAT*.

[79] Mohammad Nur Nobi et al. 2022. Toward Deep Learning Based Access Control. In *ACM CODASPY*.

[80] Mohammad Nur Nobi, Ram Krishnan, Yufei Huang, and Ravi Sandhu. 2022. Administration of Machine Learning Based Access Control. In *European Symposium on Research in Computer Security (ESORICS)*. Springer.

[81] Mohammad Nur Nobi, Ram Krishnan, and Ravi Sandhu. 2022. Adversarial Attacks in Machine Learning Based Access Control. In *Italian Conference on Big Data and Data Science (ITADATA)*. CEUR.

[82] Aissam Outchakoucht, ES Hamza, and Jean Philippe Leroy. 2017. Dynamic access control policy based on blockchain and machine learning for the internet of things. *Int. J. Adv. Comput. Sci. Appl* (2017).

[83] Seth Proctor. 2005. Sun XACML Implementation. http://sunxacml.sourceforge.net/

[84] Ravi Sandhu et al. 1999. The ARBAC99 model for administration of roles. In *IEEE ACSAC*.

[85] Daniel Servos et al. 2017. Current research and open problems in attribute-based access control. *ACM CSUR* (2017).

[86] Mahendra Pratap Singh et al. 2021. A Role-Based Administrative Model for Administration of Heterogeneous Access Control Policies and its Security Analysis. *Information Systems Frontiers* (2021).

[87] John Slankas et al. 2013. Access control policy identification and extraction from project documentation. *SCIENCE*.

[88] John Slankas, Xusheng Xiao, Laurie Williams, and Tao Xie. 2014. Relation extraction for inferring access control rules from natural language artifacts. In *Proceedings of the 30th Annual Computer Security Applications Conference*.

[89] Eduardo J Spinosa et al. 2009. Novelty detection with application to data streams. *Intelligent Data Analysis* (2009).

[90] Kriti Srivastava et al. 2020. 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*. Springer.

[91] Scott D Stoller. 2015. An administrative model for relationship-based access control. In *DBSec*. Springer.

[92] Ian P Turnipseed. 2015. *A new scada dataset for intrusion detection research*. Mississippi State University.

[93] UCI. 2011. Amazon Access Data Set. http://archive.ics.uci.edu/ml/datasets/Amazon+Access+Samples

[94] Guido Urdaneta, Guillaume Pierre, and Maarten Van Steen. 2009. Wikipedia workload analysis for decentralized hosting. *Computer Networks* (2009).

[95] Richard Van De Stadt. 2012. Cyberchair: A web-based groupware application to facilitate the paper reviewing process. *arXiv* (2012).

[96] Chengcheng Xiang et al. 2019. Towards Continuous Access Control Validation and Forensics. In *ACM CCS*.

[97] Xusheng Xiao, Amit Paradkar, Suresh Thummalapenta, and Tao Xie. 2012. Automated extraction of security policies from natural-language software documents. In *ACM Symposium on the Foundations of Software Engineering*.

[98] Zhongyuan Xu and Scott D Stoller. 2014. Mining attribute-based access control policies from logs. In *DBSec*. Springer.

[99] Michalis Xyntarakis and Constantinos Antoniou. 2019. Data science and data visualization. In *Mobility Patterns, Big Data and Transport Analytics*. Elsevier.

[100] Tianhang Zheng, Changyou Chen, and Kui Ren. 2019. Distributionally adversarial attack. In *AAAI Conference on AI*.

[101] Lu Zhou, Chunhua Su, and Li et al. 2019. Automatic fine-grained access control in SCADA by machine learning. *Future Generation Computer Systems* (2019).