

CONSTRAINT-GUIDED BACKTRACKING FOR INTERPRETABLE FIREWALL DECISION ANALYSIS USING REAL UNIVERSITY NETWORK LOGS

Temilade Fasina, Agnes Akinwole, Mayowa Idowu, Sakpere Wilson

Author and Correspondence Author, Co- Author, Co-Author, Co-Author

Department of Computer Science,
Lead City University, Ibadan, Oyo State, Nigeria

Abstract : Backtracking is an exact search strategy whose practical effectiveness depends on whether constraints can eliminate unproductive branches before a complete candidate is formed. This study evaluates constraint-guided backtracking on a real and independently verifiable firewall-log dataset and examines whether exact rule search can provide a transparent alternative to purely predictive classification. The study used a UCI Internet Firewall Data contain 65,532 traffic records and 12 variables. After 8,362 exact duplicates were removed before splitting, 57,170 unique observations remained. Nine interpretable categorical features were constructed, and backtracking searched conjunctions of at most four predicates under minimum-support and minimum-precision constraints. Support pruning reduced the candidate partial-rule space from 55,412 states to 24,993 visited states, representing a 54.90% reduction. The resulting ordered rule list achieved 99.09% accuracy and a weighted F1 score of 0.9907 on 17,151 held-out records. A depth-3 decision tree achieved 99.29% accuracy and a weighted F1 score of 0.9926. The findings show that backtracking can recover compact and auditable firewall-decision patterns with predictive performance close to a shallow tree.

IndexTerms - Backtracking; Firewall Logs; Exact Search; Interpretable Rules; Cybersecurity; UCI Internet Firewall Data

1. INTRODUCTION

Firewalls are central enforcement points in network security because they translate security policy into decisions about whether traffic should be permitted, rejected, silently dropped, or reset. Modern firewall research increasingly treats these controls as both policy mechanisms and analyzable security systems, particularly where rule refinement, explainability, and testing must be understandable to administrators (Lee et al., 2024; Venčkauskas et al., 2026). The operational logs generated by these devices provide evidence of how policy decisions are actually applied to network sessions, but the volume and diversity of those records make manual analysis increasingly difficult.

Studies have therefore relied heavily on machine learning, anomaly detection, and data-driven classification. Rahman et al. (2023) applied multiple classifiers to multiclass internet firewall logs and reported 99% accuracy with random forest, while Korkmaz et al. (2024) examined firewall packet classification under severe class imbalance using artificial neural networks and SMOTE-family methods. A recent firewall-focused survey likewise shows that machine-learning and deep-learning techniques now dominate anomaly-detection research, but it also identifies continuing concerns around dataset selection, imbalance, evaluation, and operational interpretability (Dhrir et al., 2025). High predictive performance is valuable, yet a security analyst may still need to know the exact conditions that produced a decision and whether those conditions can be reproduced independently.

Backtracking provides a complementary exact-search view. It constructs candidate solutions incrementally and abandons a branch as soon as an explicit constraint proves that no acceptable descendant can exist. Recent exact-search research continues to rely on structured extension, feasibility testing, and branch rejection as the central mechanisms that make combinatorial search practical within a declared model space (Ordyniak et al., 2024; Seidel et al., 2026). In a firewall-log setting, a candidate can be defined as a conjunction of human-readable predicates such as a destination-port category, a NAT-state relation, an elapsed-time category, or a traffic-volume band. The search can then be constrained by minimum support, minimum class precision, and maximum rule length. This study addresses three questions: (1) Can exact backtracking discover compact, high-precision firewall-action rules from real operational logs? (2) To what extent can mathematically safe pruning reduce the search effort? and (3) How does a small ordered rule list compare with an interpretable depth-3 decision-tree baseline on held-out records? These questions separate predictive performance from explanatory transparency. Research on interpretable rule learning similarly treats rule size, computational tractability, and predictive quality as simultaneous design concerns rather than assuming that accuracy alone defines a useful model (Pellegrina & Vandin, 2024; Röber et al., 2025).

2. LITERATURE REVIEW AND RELATED WORKS

2.1 Firewall logs as a cybersecurity evidence source

Recent studies treat firewall logs as a substantive source of cybersecurity evidence rather than merely as troubleshooting records. Rahman et al. (2023) used multiclass firewall logs to learn the operational actions assigned to network sessions, demonstrating that firewall decisions contain strong and learnable structure. Andalib and Babamir (2023) analyzed large distributed-firewall logs in a real security environment and showed that log-derived patterns can support the detection of firewall-policy anomalies. These studies establish two complementary uses of telemetry: prediction of operational decisions and retrospective examination of policy behavior.

Komadina et al. (2024) analyzed firewall logs from a large industrial control network and compared supervised and unsupervised anomaly-detection approaches after integrating realistic attack-oriented anomalies. Lee et al. (2024) examined firewall rule refinement in a high-performance computing service network, linking automated analysis directly to policy maintenance. Taken together, this literature supports the use of real firewall telemetry for security analysis while also showing that the strongest contributions connect data patterns to operationally meaningful security decisions (Andalib & Babamir, 2023; Komadina et al., 2024; Lee et al., 2024).

2.2 Machine-learning classification of firewall actions

Rahman et al. (2023) provide one of the most directly comparable recent studies of the four-action firewall classification problem. They evaluated several machine-learning algorithms on internet firewall logs and reported 99% accuracy for random forest while also examining F1 score, recall, and sensitivity. Their findings confirm that allow, deny, drop, and reset-both decisions can be learned with very high aggregate accuracy. The result also creates a demanding baseline for any new method: contribution cannot be justified merely by reproducing a high overall score on a highly structured dataset.

Korkmaz et al. (2024) addressed the same general firewall-packet classification problem from the perspective of class imbalance. They combined artificial neural networks with SMOTE, ADASYN, and BorderlineSMOTE and reported that balancing strategies materially affected minority-class behavior. This is particularly relevant because reset-both is extremely rare in the UCI firewall data. Their study shows why overall accuracy should be interpreted alongside macro, balanced, and class-specific measures, especially when a dominant class can mask poor performance on a security-relevant minority class.

Dhrir et al. (2025) reviewed machine-learning and deep-learning approaches specifically for firewall anomaly detection and identified continuing challenges involving datasets, evaluation metrics, model choice, and operational complexity. Their synthesis supports a broader interpretation of firewall analytics: predictive power remains important, but reproducibility, class balance, explainability, and deployment context increasingly determine whether a method is useful in practice. The present study therefore treats exact rule search as a complementary analytic approach, not as a replacement for high-capacity predictive models (Dhrir et al., 2025).

2.3 Firewall rule refinement, formal logic, and explainability

Lee et al. (2024) proposed an AI-based approach to firewall rule refinement in a high-performance-computing service network, demonstrating the practical need to identify unnecessary or problematic rules without losing administrative interpretability. Elfaki et al. (2026) developed a logic-driven firewall anomaly-detection framework supported by knowledge-graph visualization and machine-learning validation. Venčkauskas et al. (2026) likewise made explainability a central component of firewall penetration testing. These studies show a clear recent movement toward security methods that expose the reasoning behind detected or recommended firewall behavior.

Dhrir et al. (2025) report increasing use of advanced machine-learning and deep-learning techniques but also emphasize unresolved challenges around model evaluation, datasets, and practical deployment. The recent logic-driven approach of Elfaki et al. (2026) is especially relevant because it treats deterministic, human-inspectable security conditions as part of the detection mechanism itself. Constraint-guided backtracking follows the same design principle at a smaller scale: a decision is represented by an explicit conjunction whose support, precision, and search history can be inspected directly.

2.4 Backtracking, constraint satisfaction, and exact interpretable rule learning

Ordyniak et al. (2024) developed a general framework for learning smallest interpretable symbolic models, including decision lists, decision sets, decision trees, and binary decision diagrams, showing that compact symbolic learning can be analyzed through explicit algorithmic structure. Recent rule-list research further clarifies the relationship between interpretability and search complexity. Pellegrina and Vandin (2024) show that learning high-quality rule lists can be computationally difficult and propose a scalable sampling method with explicit quality guarantees. Röber et al. (2025) formulate rule generation as a constrained optimization problem that addresses scalability, interpretability, and fairness. In the present study, the rule language is deliberately narrower and the pruning condition is simpler, but the underlying motivation is shared: model transparency must be built into the search objective and computational strategy rather than added only after prediction (Pellegrina & Vandin, 2024; Röber et al., 2025).

The literature reveals three gaps. First, current firewall-log studies continue to emphasize classifier performance and imbalance handling, with less attention to exhaustive discovery of short, auditable conjunctions within a fully declared search space (Korkmaz et al., 2024; Rahman et al., 2023). Second, recent policy-oriented research increasingly values logic, explainability, and administrator-facing rule refinement, but it typically does not connect those goals to an exact data-derived rule search over operational firewall logs (Elfaki et al., 2026; Lee et al., 2024). Third interpretable-learning research recognizes the computational difficulty of finding compact rule models and therefore relies on optimization, sampling, or exact symbolic frameworks (Ordyniak et al., 2024; Pellegrina & Vandin, 2024; Röber et al., 2025). The research gap is therefore a transparent and reproducible account of how a restricted firewall rule space can be searched exactly, pruned safely, and evaluated against a conventional interpretable baseline.

3. THEORETICAL FRAMEWORK OF CONSTRAINT-GUIDED BACKTRACKING

3.1 Search-space representation

Let the training data be $D = \{(x_i, y_i)\}$ for $i = 1, \dots, n$, where x_i is a firewall record and y_i belongs to the action set $A = \{\text{allow, deny, drop, reset-both}\}$. Let $F = \{f_1, \dots, f_m\}$ denote the m derived categorical features. Each feature f_j has a finite category set V_j . A primitive predicate has the form $p = (f_j = v)$, where v belongs to V_j . A candidate firewall rule R is a conjunction of predicates drawn from distinct features.

$$R = \bigwedge_{\ell=1}^d p_{\ell}, \quad 1 \leq d \leq K \quad (1)$$

The restriction to distinct features avoids contradictory conditions such as assigning two different categories to the same derived feature. K is the maximum rule length. In this study $K = 4$. A candidate rule defines a matching set $M(R)$ containing all training records that satisfy every predicate in R . This finite symbolic representation is consistent with recent exact-interpretable-model research in which compact decision lists, decision sets, trees, and related symbolic structures are learned by explicitly searching constrained model spaces (Ordyniak et al., 2024; Pellegrina & Vandin, 2024).

$$M(R) = \{i \in \{1, \dots, n\} : R(x_i) = 1\} \quad (2)$$

For a target action a , let N_a be the number of training observations with class a . Let $TP_a(R)$ be the number of matched records whose action is a . The basic descriptive quantities of a rule are support, precision, and recall. For compact notation in the equations, $P_a(R)$ denotes precision and $Q_a(R)$ denotes recall.

$$\text{supp}(R) = |M(R)| \quad (3)$$

$$P_a(R) = \frac{TP_a(R)}{|M(R)|} \quad (4)$$

$$Q_a(R) = \frac{TP_a(R)}{N_a} \quad (5)$$

3.2 Size of the unpruned rule tree

If feature f_j has c_j categories, the number of possible conjunctions of exactly d predicates, when at most one predicate may be selected from each feature, is obtained by summing the products of category counts over all d -feature subsets. This gives a finite but rapidly expanding state space.

$$N_d = \sum_{S \subseteq \{1, \dots, m\} \atop |S|=d} \prod_{j \in S} c_j \quad (6)$$

$$N_{\text{total}}(K) = \sum_{d=1}^K N_d \quad (7)$$

For the nine categorical features used in this study, the observed category structure yields 55,412 candidate partial conjunctions across depths one through four. Rule-learning studies similarly identify scalability as a central obstacle when the number of candidate rules grows combinatorially, motivating sampling, column generation, and other search-space controls (Pellegrina & Vandin, 2024; Röber et al., 2025).

3.3 Anti-monotone support and safe pruning

The central pruning rule follows from a simple set relation. Adding a predicate to a conjunction can only keep or reduce the records that match it. Suppose R is a partial rule and q is another predicate. Because every record satisfying $R \wedge q$ must also satisfy R , the matching set of the extension is a subset of the matching set of its parent.

$$M(R \wedge q) \subseteq M(R) \quad (8)$$

$$\text{supp}(R \wedge q) \leq \text{supp}(R) \quad (9)$$

Equation 9 is anti-monotonicity of support. It produces a safe pruning theorem: if $\text{support}(R) < s_{\min}$, then no descendant formed by adding one or more predicates can satisfy the minimum-support requirement. The algorithm can therefore stop exploring that branch without risking the loss of a feasible rule.

3.4 General recursive backtracking algorithm

The search is depth-first. At each node, the algorithm chooses a feature not already present in the rule, tries each observed category, intersects the parent match mask with the predicate mask, and evaluates the resulting support. The recursive path is abandoned immediately when support falls below $s_{\min}$. Otherwise, the candidate is evaluated for precision and, if the maximum depth has not been reached, extended recursively.

Algorithm 1: SUPPORT-PRUNED-BACKTRACK(target_action a)

```

1 SEARCH(start_feature = 1, current_mask = ALL_RECORDS, conditions = empty)
2   for each unused feature  $f$  from start_feature onward
3     for each observed category  $v$  of  $f$ 
4       new_mask = current_mask AND mask( $f = v$ )
5       if count(new_mask) <  $s_{\min}$ : continue # safe prune
6        $R = \text{conditions} + (f = v)$ 
7       compute precision_a( $R$ ) and recall_a( $R$ )
8       if precision_a( $R$ ) >=  $p_{\min}$ : store  $R$ 
9       if length( $R$ ) <  $K$ : SEARCH(next feature, new_mask,  $R$ )
10 return the shortest qualifying rule; break ties by recall then precision

```

4. MATERIALS AND METHODS

4.1 Dataset source and independent verification

The empirical analysis uses the Internet Firewall Data from the UCI Machine Learning Repository (UCI Machine Learning Repository, 2019). UCI reports that the data were collected from internet traffic records on a university firewall. The repository lists 65,532 instances, 12 variables, no missing values, and the Action variable as the class field. Recent work continues to use the allow, deny, drop, and reset-both decision structure for multiclass firewall-log and packet classification, confirming the dataset's relevance as a reproducible firewall benchmark (Korkmaz et al., 2024; Rahman et al., 2023). The dataset is distributed as log2.csv under the Creative Commons Attribution 4.0 International licence and has the persistent DOI <https://doi.org/10.24432/C5131M>. The exact CSV used was SHA-256 digest is 0b42e7eb9a4d7c314f65810c447ebf7f09d5e3e106ec14d36950a9ebac61e9c0. The row count, column names, missing-value count, and observed class labels were checked against the UCI metadata and the raw file. The cryptographic digest provides a file-level identifier so that another researcher can confirm that the same input data were used even if the repository interface changes.

4.2 Duplicate control and data partitioning

The raw file contained 8,362 exact duplicate rows. These were removed before any training-test split, leaving 57,170 unique observations. Removing duplicates first prevents an identical record from appearing in both partitions and producing an overly optimistic held-out estimate. Recent data-centric cybersecurity research likewise treats duplicate removal, explicit preprocessing, and reproducible dataset construction as part of technical validation rather than as undocumented cleaning steps (Quirumbay Yagual et al., 2026). The unique records were then divided using a stratified 70:30 split with random seed 42, yielding 40,019 training observations and 17,151 test observations.

Table 2. Distribution of firewall actions before and after duplicate removal.

Action	Raw n	Raw %	Unique n	Unique %
Allow	37,640	57.438%	37,439	65.487%
Deny	14,987	22.870%	8,042	14.067%
Drop	12,851	19.610%	11,635	20.352%
reset-both	54	0.082%	54	0.094%

4.3 Interpretable predicate construction

Nine categorical features were derived from the original numerical columns to keep the rule language small and understandable. Source and destination ports were grouped using the IANA System, User, and Dynamic/Private ranges, while seven commonly occurring destination-port values were retained explicitly. NAT source and NAT destination states indicate whether the translated port is zero, unchanged, or translated. Bytes, packet counts, elapsed time, received packets, and received bytes were grouped into fixed magnitude bands. This construction reflects the recent emphasis on firewall representations that remain intelligible to administrators and can be connected to explicit security logic (Elfaki et al., 2026; Lee et al., 2024). These transformations were declared in code and were not optimized against the test set.

Table 3. Derived categorical features used by the exact search.

Feature	Categories
Source-port class	system; user; dynamic
Destination service	53, 80, 123, 443, 445, 3389, 5900; other system/user/dynamic
NAT source state	zero; unchanged; translated
NAT destination state	zero; unchanged; translated
Bytes band	<100; 100-999; 1k-9.9k; 10k-99.9k; >=100k
Packets band	1; 2-9; 10-99; >=100
Elapsed-time band	0; 1-9; 10-59; 60-599; >=600 s
Received-packet band	0; 1; 2-9; >=10
Received-byte band	0; 1-99; 100-999; >=1000

4.4 Exact-search parameters and rule selection

The maximum conjunction length was four predicates. Minimum support was five training records, and a qualifying rule required at least 98% training precision. The low support threshold was necessary because reset-both contains only 54 records in the entire raw dataset. For each class, the shortest qualifying rule was selected; ties were resolved by higher training recall and then higher training precision. Limiting model size is consistent with recent interpretable-learning research in which compactness is an explicit design goal and computational constraint (Ordyniak et al., 2024; Pellegrina & Vandin, 2024).

4.5 Baseline model and evaluation metrics

A DecisionTreeClassifier with maximum depth three was fitted to the 11 original numeric predictors using the same training and test records. The shallow depth was chosen because it provides an interpretable baseline rather than an unrestricted high-capacity tree. Evaluation uses accuracy, macro precision, macro recall, macro F1, weighted F1, and balanced accuracy. Recent firewall classification research shows why this broader metric set is necessary when minority actions are highly underrepresented (Korkmaz et al., 2024), and the firewall anomaly-detection literature similarly emphasizes careful metric and dataset selection when comparing models (Dhrir et al., 2025).

5. RESULTS

5.1 Dataset integrity and class imbalance

The raw data contain 37,640 allow records, 14,987 deny records, 12,851 drop records, and 54 reset-both records. After duplicate removal, the unique counts are 37,439 allow, 8,042 deny, 11,635 drop, and 54 reset-both. The distribution is therefore highly imbalanced, particularly for reset-both. This pattern is important because recent firewall classification research demonstrates that minority-class handling can materially change model behavior even when aggregate accuracy remains high (Korkmaz et al., 2024), while recent surveys identify imbalance and dataset characteristics as recurring evaluation concerns in firewall anomaly detection (Dhrir et al., 2025).

5.2 Search-space reduction

The nine categorical features generated 41 primitive category predicates. Across rule depths one to four, the corresponding unpruned search tree contains 55,412 possible partial conjunctions. Minimum-support pruning reduced this to 24,993 visited states and avoided 30,419 states, a reduction of 54.90%. Because the prune is based on the anti-monotone support property established in Equation 9, the reduction does not sacrifice any candidate that could later recover the required minimum support. The result is consistent with current exact symbolic learning research, where search-space control is central to making interpretable model discovery computationally feasible (Ordyniak et al., 2024; Pellegrina & Vandin, 2024).

Table 4. Backtracking search-space accounting.

Measure	Observed value
Primitive predicate categories	41
Maximum rule length	4
Minimum support	5
Minimum training precision	98%
Unpruned partial conjunctions	55,412
Visited after support pruning	24,993
Avoided states	30,419
Search reduction	54.90%

5.3 Selected action rules

Table 5. Exact-search rules and held-out performance.

Action	Selected rule	Train precision	Train recall	Test precision	Test recall
Allow	NAT destination port = destination port	99.93%	98.88%	99.94%	98.88%
Deny	other user-range destination port AND elapsed time = 0 s	99.31%	60.93%	99.26%	60.84%
Drop	destination port = 445	99.67%	100.00%	99.66%	100.00%
reset-both	destination port = 3389 AND bytes = 100-999	100.00%	26.32%	100.00%	12.50%

The allow rule generalized strongly: an unchanged NAT destination port achieved 99.94% test precision and 98.88% test recall. Destination port 445 captured all held-out drop records with 99.66% precision. The deny rule remained highly precise at 99.26% but covered 60.84% of held-out deny observations, indicating greater heterogeneity. The reset-both rule retained 100% precision but detected only 2 of 16 held-out reset-both records, producing 12.5% recall. The direct readability of these conditions is aligned with recent firewall work that emphasizes rule refinement, formal logic, and explainable security decisions (Elfaki et al., 2026; Lee et al., 2024; Venčkauskas et al., 2026).

5.4 Multiclass performance and baseline comparison

Table 6. Held-out performance on 17,151 unique firewall records.

Model	Accuracy	Macro precision	Macro recall	Macro F1	Weighted F1	Balanced accuracy
Backtracking rule list	99.09%	0.9853	0.7768	0.7958	0.9907	0.7768
Depth-3 decision tree	99.29%	0.9883	0.7778	0.7979	0.9926	0.7778

The shallow decision tree achieved 99.29% accuracy, compared with 99.09% for the ordered backtracking rule list. Weighted F1 values were 0.9926 and 0.9907, respectively. Macro F1 was 0.7979 for the tree and 0.7958 for the rule list, while balanced accuracy was 0.7778 and 0.7768. The result therefore does not support a claim that exact backtracking is more accurate than a shallow tree. Instead, it shows that four explicitly constrained rules can approach the predictive performance of an interpretable tree on this dataset. This high overall predictability is consistent with firewall-log studies in which conventional machine-learning and neural approaches also achieve strong multiclass results (Korkmaz et al., 2024; Rahman et al., 2023).

Table 7. Confusion matrix for the backtracking rule list.

Actual / Predicted	Allow	Deny	Drop	Reset-both
Allow	11106	126	0	0
Deny	4	2397	12	0
Drop	0	0	3490	0
Reset-both	0	14	0	2

6. DISCUSSION OF FINDINGS

6.1 Exact rule search and predictive performance

The first finding is that a deliberately small exact rule language can reproduce most firewall actions in the held-out UCI records. The 99.09% accuracy of the rule list is close to the 99.29% depth-3 decision-tree result. Rahman et al. (2023) reported 99% accuracy with random forest on multiclass internet firewall logs, confirming that the four firewall actions contain highly learnable structure. Korkmaz et al. (2024) also obtained strong multiclass firewall-packet performance while focusing specifically on class balancing. The present result therefore falls within the high-performance range observed in recent firewall classification research, but its contribution lies in producing a compact exact rule list and an auditable search path rather than a claim of predictive superiority (Dhrir et al., 2025).

The marginal advantage of the shallow decision tree is scientifically useful because it prevents the contribution from being overstated. Research shows that compact symbolic models involve explicit trade-offs among accuracy, model size, and computational effort (Ordyniak et al., 2024; Pellegrina & Vandin, 2024; Röber et al., 2025). In this study, the backtracking framework is therefore justified by transparency of the feasibility constraints, compactness of the selected rules, and reproducibility of the search process, not by a small difference in aggregate predictive score.

6.2 Search pruning and classical constraint-satisfaction theory

The second finding is the 54.90% reduction in visited rule states created by the support prune. This result is consistent with recent exact symbolic-model research in which search becomes practical only when provably irrelevant or dominated regions of the candidate space can be excluded (Ordyniak et al., 2024). Pellegrina and Vandin (2024) likewise identify computational scalability as a central challenge in rule-list learning, while Seidel et al. (2026) retain explicit accept and reject mechanisms in a modern implementation of general backtracking. The firewall result provides a concrete empirical example: once support falls below the threshold, further specialization cannot restore it, so the branch can be removed safely. The finding also clarifies the difference between exactness and exhaustive enumeration without pruning. Exact-search systems do not need to evaluate every complete model if mathematically valid bounds, rejection tests, or optimization structures exclude regions that cannot contain an acceptable solution. Recent rule-generation research uses related ideas to control scalability through explicit optimization and search structure (Röber et al., 2025). In the present experiment, the support bound is intentionally simple, but it eliminated more than half of all candidate partial conjunctions while preserving every descendant that could still satisfy the declared minimum support.

6.3 Interpretability as part of the decision mechanism

The selected rules demonstrate intrinsic interpretability because a prediction can be traced directly to the condition that produced it. An allow decision can be associated with NAT destination port = destination port, while drop can be associated with destination port 445. This design aligns closely with recent firewall-security research. Lee et al. (2024) connect AI analysis to firewall rule refinement, Elfaki et al. (2026) use formal logic and knowledge-graph visualization to make firewall anomaly decisions inspectable, and Venčkauskas et al. (2026) explicitly integrate explainability into machine-learning-based firewall penetration testing. The present framework follows the same operational principle: the explanation is embedded in the decision representation rather than generated only after prediction.

6.4 Class imbalance and the reset-both limitation

The major weakness is the reset-both class. Only 54 raw observations exist, and the held-out test set contains 16 unique reset-both records. The selected rule identified only two of them, producing 12.5% recall even though its precision was 100%. The depth-3 decision tree had the same reset-both recall. This explains why overall accuracy is near 99% while balanced accuracy remains around 0.78. Studies confirm that class imbalance can materially distort multiclass evaluation and motivates class-specific analysis rather than reliance on overall accuracy alone (Dhrir et al., 2025; Korkmaz et al., 2024). Korkmaz et al. (2024) approached firewall class imbalance by combining artificial neural networks with SMOTE-family techniques and found that balancing strategies improved minority-class treatment. Their result indicates that oversampling can be valuable when minority detection is the primary goal. The present analysis deliberately does not oversample because the objective is to discover rules supported by the observed firewall records. Consequently, the poor reset-both recall is an empirical boundary of the current rule-search design rather than a result to be hidden. Future work should compare original-data search, training-only oversampling, and cost-sensitive rule selection under the same evaluation protocol (Korkmaz et al., 2024).

6.5 Relationship to real firewall-policy and log-anomaly studies

The practical meaning of the results is supported by studies using operational firewall environments rather than only benchmark prediction. Andalib and Babamir (2023) extracted patterns from large distributed-firewall logs to detect policy anomalies. Komadina et al. (2024) analyzed firewall logs from a large industrial control network and compared detection strategies in an operationally grounded setting. Lee et al. (2024) addressed AI-supported firewall rule refinement in a high-performance-computing service network. The present method contributes a different mechanism to this line of work by producing short conjunctions with declared support and precision requirements. Studies have strengthened this interpretation. Elfaki et al. (2026) propose deterministic and explainable firewall anomaly definitions using formal logic and knowledge-graph visualization, while Venčkauskas et al. (2026) incorporate explainability into firewall penetration testing. The backtracking framework is less expressive than a full first-order logic or penetration-testing architecture, but it shares the principle that transparency should be built into the analytic representation. In security policy analysis, a technically accurate result is more actionable when an administrator can inspect the conditions and evidence supporting it.

7. PRACTICAL IMPLICATIONS

The method is best positioned as a security-analysis and audit aid. An analyst can define a restricted predicate language, declare minimum support and precision requirements, run an exhaustive search within that language, and receive compact rules with transparent evidence counts. This use is consistent with firewall research that connects automated analysis to rule refinement, explainable anomaly reasoning, and analyst-facing security assessment (Elfaki et al., 2026; Lee et al., 2024; Venčkauskas et al., 2026).

8. LIMITATIONS AND FUTURE RESEARCH

One major limitation is class imbalance. Reset-both is too rare for a simple minimum-support, high-precision objective to produce broad coverage. SMOTE-family balancing strategies can materially alter minority-class performance in firewall packet classification. Future work should therefore examine class-specific support thresholds, cost-sensitive objectives, training-only oversampling, and multiobjective selection while preserving a clear distinction between logically safe pruning conditions and heuristics that only guide the search. Another limitation is search-space restriction. Exactness applies only to the declared categorical predicate library and maximum rule length of four. Extending the language to arbitrary numerical thresholds, IP prefixes, protocols, temporal sequences, or rule interactions would enlarge the state space considerably. Stronger bounds, memoization, branch-and-bound objectives, constraint propagation, or SAT/CP-SAT encodings should therefore be evaluated for richer firewall rule languages.

9. CONCLUSION

This study demonstrates a constraint-guided backtracking framework for interpretable firewall decision analysis using real and verifiable university network logs. After exact duplicate removal, 57,170 unique UCI firewall records were analyzed. Anti-monotone support pruning reduced the candidate partial-rule space by 54.90%, and the selected four-rule list achieved 99.09% accuracy on held-out data. The strong predictive structure is consistent with recent studies, while the emphasis on compact symbolic decisions aligns with current work on interpretable model learning and scalable rule lists.

References

- Andalib, A., & Babamir, S. M. (2023). Anomaly detection of policies in distributed firewalls using data log analysis. *The Journal of Supercomputing*, 79, 19473–19514. <https://doi.org/10.1007/s11227-023-05417-7>
- Dhrir, H., Charfeddine, M., & Kammoun, H. M. (2025). Machine learning- and deep learning-based anomaly detection in firewalls: A survey. *The Journal of Supercomputing*, 81(6), Article 761. <https://doi.org/10.1007/s11227-025-07212-y>
- Elfaki, A. O., Albluwi, A., Aljaedi, A., & Nerma, M. H. M. (2026). Explainable logic-driven firewall anomaly detection with knowledge graph visualization and machine learning validation. *Electronics*, 15(8), 1714. <https://doi.org/10.3390/electronics15081714>
- Komadina, A., Kovačević, I., Štengl, B., & Groš, S. (2024). Comparative analysis of anomaly detection approaches in firewall logs: Integrating light-weight synthesis of security logs and artificially generated attack detection. *Sensors*, 24(8), 2636. <https://doi.org/10.3390/s24082636>
- Korkmaz, A., Bulut, S., Talan, T., Kosunalp, S., & Iliev, T. (2024). Enhancing firewall packet classification through artificial neural networks and synthetic minority over-sampling technique: An innovative approach with evaluative comparison. *Applied Sciences*, 14(16), 7426. <https://doi.org/10.3390/app14167426>
- Lee, J.-K., Hong, T., & Lee, G. (2024). AI-based approach to firewall rule refinement on high-performance computing service network. *Applied Sciences*, 14(11), 4373. <https://doi.org/10.3390/app14114373>
- Ordyniak, S., Paesani, G., Rychlicki, M., & Szeider, S. (2024). A general theoretical framework for learning smallest interpretable models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(9), 10662–10669. <https://doi.org/10.1609/aaai.v38i9.28937>
- Pellegrina, L., & Vandin, F. (2024). Scalable rule lists learning with sampling. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 2352–2363). Association for Computing Machinery. <https://doi.org/10.1145/3637528.3671989>
- Quirumbay Yagual, D., Fernández Iglesias, D., Nóvoa, F. J., & Garabato, D. (2026). A data-centric network traffic dataset for anomaly detection: Construction, reproducible pipeline, and technical validation. *Data*, 11(8), 199. <https://doi.org/10.3390/data11080199>
- Rahman, M. H., Islam, T., Rana, M. M., Tasnim, R., Mona, T. R., & Sakib, M. M. (2023). Machine learning approach on multiclass classification of internet firewall log files. In *2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES)* (pp. 358–364). IEEE. <https://doi.org/10.1109/CISES58720.2023.10183601>
- Röber, T. E., Lumadjeng, A. C., Akyüz, M. H., & Birbil, Ş. İ. (2025). Rule generation for classification: Scalability, interpretability, and fairness. *Computers & Operations Research*, 183, 107163. <https://doi.org/10.1016/j.cor.2025.107163>
- Seidel, R., Zander, R., Petrič, M., Steinmann, N., Liu, D. Q., Tcholtchev, N., & Hauswirth, M. (2026). Quantum backtracking in Qrisp applied to Sudoku problems. *ACM Transactions on Quantum Computing*, 7(3), Article 20, 1–32. <https://doi.org/10.1145/3799888>
- UCI Machine Learning Repository. (2019). *Internet Firewall Data* [Data set]. <https://doi.org/10.24432/C5131M>
- Venčkauskas, A., Toldinas, J., & Morkevičius, N. (2026). Explainable firewall penetration testing method employing machine learning. *Electronics*, 15(5), 1030. <https://doi.org/10.3390/electronics15051030>

Copyright & License:



© Authors retain the copyright of this article. This work is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.