Patent Application: Predictive Policy Compliance Engine

Title of Invention

Machine Learning System and Method for Temporal Predictive Cloud Policy Compliance Analysis with Configuration Drift Detection and Automated Risk-Based Remediation Generation

Technical Field

This invention relates to machine learning systems for cloud computing compliance management, specifically to predictive analytics engines that forecast policy compliance violations before occurrence through temporal pattern analysis, configuration drift detection, and automated generation of risk-prioritized remediation strategies.

Independent Claims

Claim 1 (System Claim - Broadest)

A computer-implemented predictive compliance system for cloud environments comprising:

a) **a compliance data collection and preprocessing engine** configured to:

- ingest policy violation events from cloud governance APIs at rates exceeding 100,000 events per minute,
- extract temporal features including time-based aggregations, seasonal patterns, and lag variables across 1-hour to 30-day windows,
- normalize heterogeneous compliance data using domain-specific encoders for policy types, resource categories, and violation severities,
- implement adaptive sampling strategies balancing data completeness with system performance;

b) **a configuration drift detection system** implementing:

- multi-dimensional state space modeling of resource configurations using variational autoencoders (VAEs),
- statistical process control with dynamic control limits adjusted for legitimate configuration changes,
- drift velocity calculation measuring rate of movement toward non-compliant states,
- anomaly scoring using Mahalanobis distance in configuration feature space;

c) **a temporal pattern analysis engine** configured to:

- decompose compliance time series into trend, seasonal, cyclical, and irregular components using STL decomposition,
- identify recurrent violation patterns using motif discovery algorithms,
- detect regime changes in compliance behavior using hidden Markov models,
- calculate temporal dependencies using transfer entropy and Granger causality;

d) **an ensemble prediction system** implementing:

- gradient boosting machines (XGBoost) for non-linear pattern recognition with custom loss functions for compliance prediction,
- long short-term memory (LSTM) networks with attention mechanisms for sequence modeling,
- Prophet models for business-hour and seasonal compliance patterns,
- Bayesian model averaging for uncertainty quantification;

e) **a risk assessment and prioritization module** configured to:

- calculate violation probability using calibrated probability estimates from ensemble models,
- assess business impact through integration with asset criticality databases,
- generate risk scores using fuzzy logic systems combining probability and impact,
- implement dynamic thresholding based on organizational risk tolerance;

f) **an automated remediation recommendation engine** implementing:

- case-based reasoning to match predicted violations with historical remediation successes,
- constraint programming to ensure recommended actions maintain system stability,
- multi-criteria decision analysis for remediation strategy selection,
- natural language generation for human-readable remediation instructions;

wherein the system achieves at least 90% precision and 85% recall for compliance violation predictions with 24-hour lead time.

Claim 2 (Method Claim - Broadest)

A computer-implemented method for predicting cloud policy compliance violations comprising:

a) **collecting and preprocessing compliance data** by:

- establishing streaming connections to cloud provider audit logs and policy engines,
- implementing event deduplication using bloom filters with false positive rate below 0.1%,

- enriching events with contextual metadata including resource tags, ownership, and business criticality,
- storing processed data in time-series databases with automatic partitioning;

b) **detecting configuration drift** by:

- learning baseline configuration distributions using kernel density estimation,
- calculating drift metrics including Kullback-Leibler divergence and Wasserstein distance,
- identifying drift patterns using change point detection algorithms,
- correlating drift patterns with historical violation data;

c) **analyzing temporal compliance patterns** by:

- applying wavelet transforms for multi-resolution time series analysis,
- implementing dynamic time warping for pattern matching across different time scales,
- using recurrent neural networks to model sequential dependencies,
- extracting cyclical patterns using Fourier analysis;

d) **generating compliance predictions** by:

- training ensemble models on labeled historical compliance data,
- implementing online learning for model adaptation to new patterns,
- applying SMOTE for handling class imbalance in violation data,
- generating prediction intervals using conformal prediction;

e) **assessing and prioritizing risks** by:

- calculating expected loss using Monte Carlo simulation,
- applying portfolio theory for risk aggregation across resources,
- implementing value-at-risk calculations for compliance exposure,
- generating risk heatmaps with drill-down capabilities;

f) **recommending remediation actions** by:

- querying knowledge bases of proven remediation strategies,
- simulating remediation outcomes using digital twin models,
- optimizing remediation sequences to minimize business disruption,
- providing confidence scores for each recommendation.

Dependent Claims

Claim 3 (Dependent on Claim 1)

The system of claim 1, wherein the configuration drift detection system further comprises:

- incremental learning algorithms updating baselines without full retraining,
- concept drift detection distinguishing between gradual and sudden drift,
- feature importance analysis identifying primary drift contributors,
- automated drift alerting with severity classification.

Claim 4 (Dependent on Claim 1)

The system of claim 1, wherein the temporal pattern analysis engine implements:

- attention-based sequence models focusing on violation-prone time periods,
- causal discovery algorithms identifying root causes of compliance patterns,
- anomaly detection using seasonal hybrid ESD (S-H-ESD) test,
- pattern clustering for grouping similar compliance behaviors.

Claim 5 (Dependent on Claim 1)

The system of claim 1, wherein the ensemble prediction system further comprises:

- automated hyperparameter optimization using Bayesian optimization,
- model stacking with meta-learners for improved accuracy,
- adversarial validation for robust predictions,
- explainable AI components using SHAP and LIME.

Claim 6 (Dependent on Claim 2)

The method of claim 2, wherein collecting compliance data includes:

- implementing privacy-preserving techniques using differential privacy,
- applying homomorphic encryption for sensitive compliance data,
- using federated learning for multi-tenant scenarios,
- maintaining audit trails with cryptographic proof of integrity.

Claim 7 (Dependent on Claim 2)

The method of claim 2, wherein analyzing temporal patterns further comprises:

- implementing matrix profile algorithms for all-pairs similarity search,
- using topological data analysis for complex pattern identification,
- applying symbolic aggregate approximation for pattern discretization,
- detecting anomalous subsequences using discord discovery.

Claim 8 (Architecture Claim)

The system of claim 1, further comprising:

- distributed computing infrastructure using Apache Spark for parallel processing,
- GPU acceleration for deep learning model training and inference,
- model serving infrastructure with sub-100ms prediction latency,
- horizontal scaling supporting 10,000+ concurrent prediction requests.

Claim 9 (Continuous Learning Claim)

The system of claim 1, implementing:

- automated model retraining triggered by performance degradation,
- A/B testing framework for gradual model deployment,
- champion-challenger model evaluation,
- feedback loops incorporating remediation outcomes.

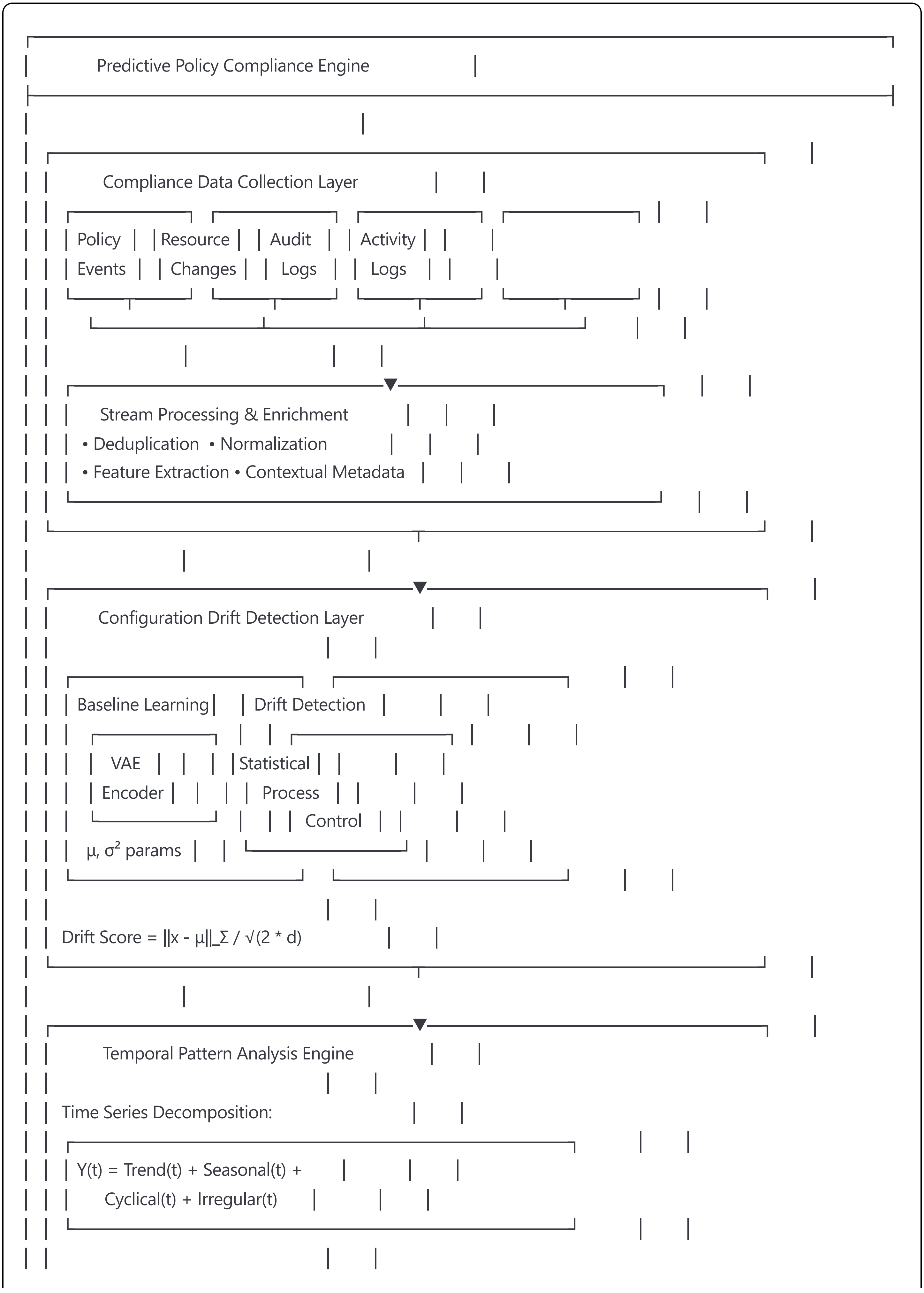
Claim 10 (Explainability Claim)

The system of claim 1, providing:

- feature contribution analysis for each prediction,
- counterfactual explanations showing paths to compliance,
- confidence calibration plots for prediction reliability,
- natural language explanations of violation risks.

Technical Diagrams

Figure 1: Predictive Compliance Engine Architecture



```
Predictive Policy Compliance Engine

Compliance Data Collection Layer

Policy      Resource    Audit    Activity
Events      Changes     Logs     Logs

Stream Processing & Enrichment
• Deduplication  • Normalization
• Feature Extraction • Contextual Metadata

Configuration Drift Detection Layer

Baseline Learning        Drift Detection

  VAE                    Statistical
  Encoder                  Process
                             Control
 μ, σ² params

Drift Score = ||x - μ||_Σ / √(2 * d)

Temporal Pattern Analysis Engine

Time Series Decomposition:

Y(t) = Trend(t) + Seasonal(t) +
     Cyclical(t) + Irregular(t)
```

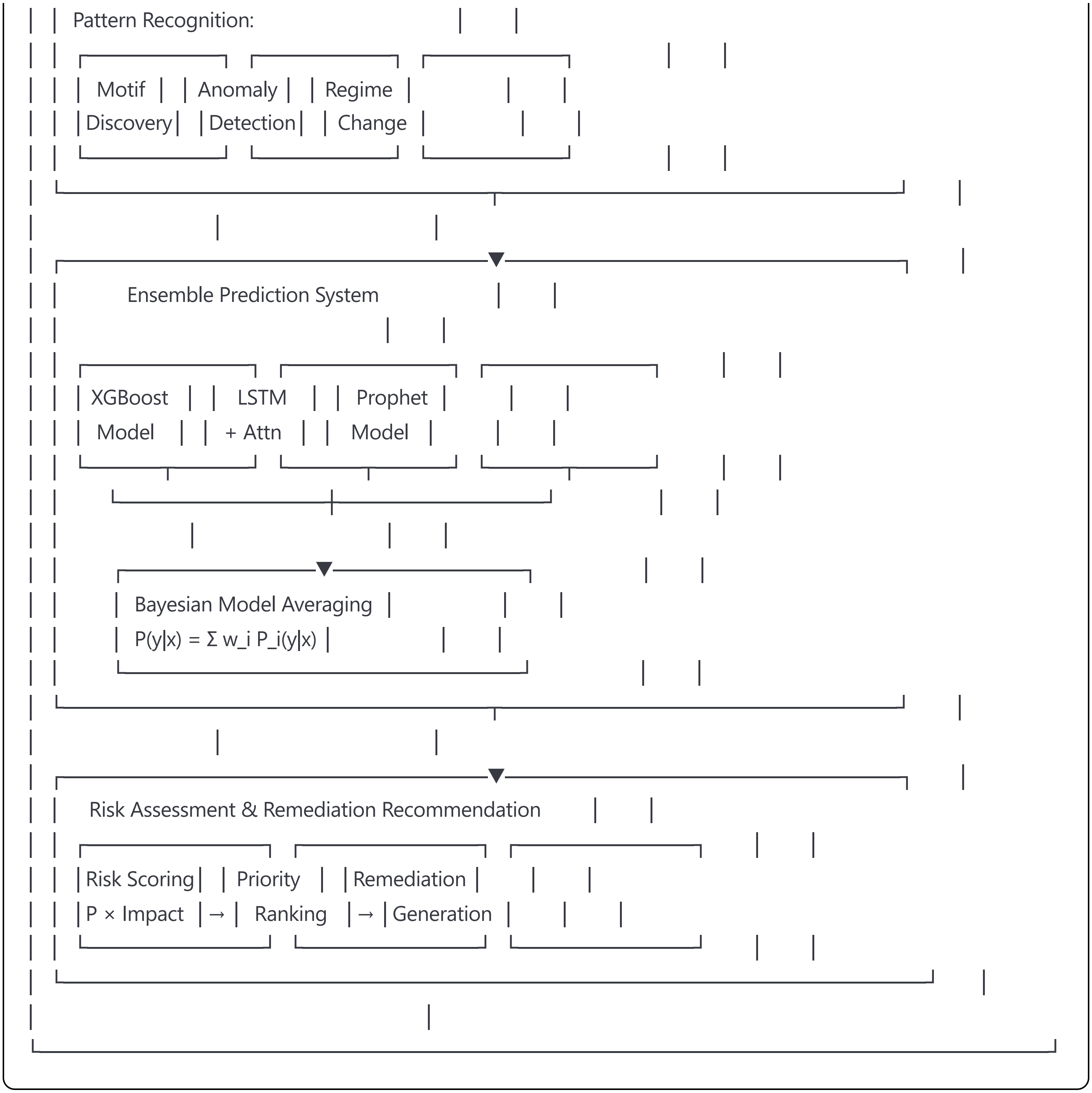


Figure 2: Configuration Drift Detection Mechanism

```
┌──────────────────────────────────────────────────────────────────────────┐
│          Configuration Drift Detection System                │
├──────────────────────────────────────────────────────────────────────────┤
│                                                  │
│ Resource Configuration Timeline                              │
│  ┌──────────────────────────────────────────────────────────┐    │
│  │ Config                                        │      │
│  │ Value   Baseline Learning    Drift Detection        │      │
│  │  ↑    ←──────────────→←──────────────────────→          │      │
│  │  │    ·········· Upper Control Limit             │      │
│  │  │  ····     ····              🚨            │      │
│  │  │ ··   ●  ●     ··  ●    ●     ●     ●          │      │
│  │  │·  ●        ●   ··  ●   ● ●    ●     ●        │      │
│  │  │··············· Mean   ●                ●       │      │
│  │  │               ················               │      │
│  │  │           Lower Control Limit ·····🚨         │      │
│  │  └──────────────────────────────────────→ Time     │      │
│  │   T₀         T₁                T_current      │      │
│  └──────────────────────────────────────────────────────────┘    │
│                                                  │
│ Variational Autoencoder Architecture                         │
│  ┌──────────────────────────────────────────────────────────┐    │
│  │                                              │      │
│  │ Input Config   Encoder       Latent   Decoder        │      │
│  │ Vector x       Network       Space    Network        │      │
│  │  ┌─────┐     ┌─────────┐   ┌───────┐   ┌─────────┐        │      │
│  │  │ x₁ │     │ Dense │   │ μ │   │ Dense │     │      │
│  │  │ x₂ │  →   │ Layers │ →  │ σ² │ → │ Layers │ →   │      │
│  │  │ .. │     │ + ReLU │   │ z~N │   │ + ReLU │     │      │
│  │  │ xₙ │     └─────────┘   └───────┘   └─────────┘        │      │
│  │  └─────┘                                   │      │
│  │                                              │      │
│  │ Loss = Reconstruction_Loss + β·KL_Divergence          │      │
│  └──────────────────────────────────────────────────────────┘    │
│                                                  │
│ Drift Detection Algorithm                                    │
│  ┌──────────────────────────────────────────────────────────┐    │
│  │ 1. Current State Encoding:                        │      │
│  │   z_current = Encoder(x_current)                  │      │
│  │                                              │      │
│  │ 2. Drift Score Calculation:                       │      │
│  │   drift_score = KL(z_current || z_baseline)           │      │
│  │                                              │      │
│  │  3. Statistical Significance Test:                      │         │
│  │     p_value = chi2_test(drift_score, df=latent_dim)     │         │
│  │                                                          │         │
│  │  4. Drift Classification:                               │         │
│  │     if p_value < 0.01: "Significant Drift"              │         │
│  │     elif p_value < 0.05: "Moderate Drift"               │         │
│  │     else: "No Significant Drift"                        │         │
│  └──────────────────────────────────────────────────────────┘         │
│                                                                       │
│  Drift Velocity & Trajectory                                          │
│  ┌──────────────────────────────────────────────────────────┐         │
│  │  Velocity = Δ(drift_score) / Δt                         │         │
│  │  Acceleration = Δ(velocity) / Δt                        │         │
│  │                                                          │         │
│  │  Predicted Time to Violation:                           │         │
│  │  t_violation = (threshold - current) / velocity         │         │
│  │                                                          │         │
│  │  Confidence Interval: [t_low, t_high] = t ± 1.96·σ_t    │         │
│  └──────────────────────────────────────────────────────────┘         │
│                                                                       │
└───────────────────────────────────────────────────────────────────────┘
```

Figure 3: Temporal Pattern Analysis

```
┌────────────────────────────────────────────────────────────────┐
│         Temporal Pattern Analysis Engine                        │
├────────────────────────────────────────────────────────────────┤
│                                                                 │
│  Multi-Resolution Time Series Analysis                          │
│  ┌──────────────────────────────────────────────────────────┐  │
│  │ Original Signal                                            │  │
│  │ ───┐  ┌──┐  ┌──┐  ┌────┐  ┌──┐  ┌───                      │  │
│  │    └──┘  └──┘  └──┘    └──┘  └──┘                           │  │
│  │                                                            │  │
│  │ Wavelet Decomposition:                                     │  │
│  │ Level 1: ∿∿∿∿∿∿∿∿∿∿∿∿∿∿∿∿ (High Freq)                       │  │
│  │ Level 2: ≈≈≈≈≈≈≈≈≈≈≈≈≈≈≈≈ (Mid Freq)                        │  │
│  │ Level 3: ──────────────── (Low Freq/Trend)                  │  │
│  └──────────────────────────────────────────────────────────┘  │
│                                                                 │
│  Pattern Recognition Pipeline                                   │
│  ┌──────────────────────────────────────────────────────────┐  │
│  │ 1. Motif Discovery (Matrix Profile)                        │  │
│  │   ┌────────────────────────────────┐                       │  │
│  │   │ Distance Matrix                 │                       │  │
│  │   │ ░▓░▓░  ← Recurring              │                       │  │
│  │   │ ░░▓░▓    Pattern                │                       │  │
│  │   │ ▓░░▓░                           │                       │  │
│  │   │ ░▓░░▓                           │                       │  │
│  │   └────────────────────────────────┘                       │  │
│  │                                                            │  │
│  │ 2. Seasonal Pattern Extraction                             │  │
│  │   Daily: Peak @ 9am, 2pm  │ Weekly: Mon/Fri spikes         │  │
│  │   Monthly: EOM processing │ Yearly: Q4 increase            │  │
│  └──────────────────────────────────────────────────────────┘  │
│                                                                 │
│  LSTM with Attention Architecture                               │
│  ┌──────────────────────────────────────────────────────────┐  │
│  │ Input Sequence: [x₁, x₂, ..., xₜ]                           │  │
│  │                                                            │  │
│  │ ┌────┐  ┌────┐  ┌────┐  ┌────┐                            │  │
│  │ │LSTM│→│LSTM│→│LSTM│→│LSTM│ Hidden States                  │  │
│  │ └─┬──┘  └─┬──┘  └─┬──┘  └─┬──┘                            │  │
│  │   h₁   h₂   h₃   hₜ                                         │  │
│  │   │    │    │    │                                          │  │
│  │ ┌─▼─────────▼─────────▼─────────▼─┐                        │  │
│  │ │  Attention Mechanism             │                        │  │
│ │ │ α = softmax(Q·K^T/√d)  │              │     │
│ │ │ Context = Σ(αᵢ·hᵢ)      │              │     │
│ │ └────────────┬──────────────┘              │     │
│ │              │                        │     │
│ │     ┌───────▼───────┐                       │     │
│ │     │Prediction │                     │     │
│ │     │ Layer    │                     │     │
│ │     └───────────┘                       │     │
│ └──────────────────────────────────────────┘     │
│                                    │
│ Causal Analysis (Granger Causality)                │
│ ┌──────────────────────────────────────────┐     │
│ │ Testing: Does X Granger-cause Y?              │     │
│ │                                    │     │
│ │ Model 1: Y_t = Σαᵢ·Y_{t-i} + ε₁              │     │
│ │ Model 2: Y_t = Σαᵢ·Y_{t-i} + Σβⱼ·X_{t-j} + ε₂    │     │
│ │                                    │     │
│ │ F-statistic = (RSS₁ - RSS₂)/p                │     │
│ │          RSS₂/(n-2p-1)                │     │
│ │                                    │     │
│ │ If F > F_critical: X Granger-causes Y          │     │
│ └──────────────────────────────────────────┘     │
│                                    │
└──────────────────────────────────────────────────┘
```

Figure 4: Ensemble Prediction System

```
┌─────────────────────────────────────────────────────────────────┐
│              Ensemble Prediction System                 │
├─────────────────────────────────────────────────────────────────┤
│                                         │
│  Feature Engineering Pipeline                           │
│  ┌───────────────────────────────────────────────────┐     │
│  │ Raw Features        Engineered Features            │     │
│  │  ┌────────────┐      ┌────────────────────┐         │     │
│  │  │ Timestamp  │  →   │ Hour, Day, Week, Month│        │     │
│  │  │ Resource   │  →   │ One-hot encoded type │        │     │
│  │  │ Policy     │  →   │ Severity, Category   │        │     │
│  │  │ Config     │  →   │ Delta from baseline  │        │     │
│  │  │ History    │  →   │ Lag-1,7,30 features  │        │     │
│  │  └────────────┘      └────────────────────┘         │     │
│  └───────────────────────────────────────────────────┘     │
│                                         │
│  Model 1: XGBoost with Custom Objective                 │
│  ┌───────────────────────────────────────────────────┐     │
│  │ Objective: Weighted Log Loss                 │     │
│  │ L = -Σ[w₊·y·log(p) + w₋·(1-y)·log(1-p)]          │     │
│  │ w₊ = 10 (violation weight), w₋ = 1               │     │
│  │                                          │     │
│  │ Tree Structure:                          │     │
│  │      [Root]                              │     │
│  │     /     \                              │     │
│  │ [f₁<0.5]  [f₁≥0.5]                        │     │
│  │   / \      / \                           │     │
│  │ 0.1  0.3  0.7  0.9  (violation probabilities)    │     │
│  └───────────────────────────────────────────────────┘     │
│                                         │
│  Model 2: LSTM with Attention                           │
│  ┌───────────────────────────────────────────────────┐     │
│  │ Architecture: 2-layer Bidirectional LSTM          │     │
│  │ Hidden Units: 128 per direction              │     │
│  │ Attention: Multi-head (8 heads)              │     │
│  │ Dropout: 0.3                             │     │
│  │ Output: Sigmoid activation               │     │
│  └───────────────────────────────────────────────────┘     │
│                                         │
│  Model 3: Prophet with Custom Seasonality               │
│  ┌───────────────────────────────────────────────────┐     │
│  │ y(t) = g(t) + s(t) + h(t) + εₜ              │     │
│  │ g(t): Piecewise linear trend              │     │
│ │ s(t): Fourier series seasonality                │       │
│ │ h(t): Holiday/maintenance effects               │       │
│ │                                                 │       │
│ │ Custom seasonalities:                           │       │
│ │ • Business hours: period=24h, fourier_order=10  │       │
│ │ • Week pattern: period=7d, fourier_order=5      │       │
│ │ • Month-end: period=30d, fourier_order=3        │       │
│ └─────────────────────────────────────────────────┘       │
│                                                           │
│ Bayesian Model Averaging                                  │
│ ┌─────────────────────────────────────────────────┐       │
│ │ Model Weights (learned):                        │       │
│ │ • XGBoost: 0.45                                 │       │
│ │ • LSTM: 0.35                                    │       │
│ │ • Prophet: 0.20                                 │       │
│ │                                                 │       │
│ │ Final Prediction:                               │       │
│ │ P(violation) = 0.45×P_xgb + 0.35×P_lstm + 0.20×P_prop │       │
│ │                                                 │       │
│ │ Uncertainty Quantification:                     │       │
│ │ σ² = Σwᵢ²σᵢ² + Σwᵢ(μᵢ - μ̄)²                      │       │
│ │ 95% CI = [P - 1.96σ, P + 1.96σ]                 │       │
│ └─────────────────────────────────────────────────┘       │
│                                                           │
└───────────────────────────────────────────────────────────┘
```

Figure 5: Risk Assessment and Remediation

```
┌─────────────────────────────────────────────────────────────────┐
│          Risk Assessment & Remediation Engine                   │
├─────────────────────────────────────────────────────────────────┤
│                                                         │
│  Risk Scoring Framework                                 │
│  ┌───────────────────────────────────────────────────┐   │
│  │ Violation Probability × Business Impact = Risk Score │   │
│  │                                                   │   │
│  │ Probability Assessment:                           │   │
│  │  ┌──────────────┐                                 │   │
│  │  │ ML Prediction│  0.73 ────┐                     │   │
│  │  │ Drift Score  │  0.81 ────┼──→ Fuzzy            │   │
│  │  │ History      │  0.65 ────┤   Aggregation       │   │
│  │  └──────────────┘           │   P = 0.78          │   │
│  │                    │                              │   │
│  │ Impact Assessment:        │                       │   │
│  │  ┌──────────────┐          │                      │   │
│  │  │ Data Class   │ Critical │                      │   │
│  │  │ User Count   │ 10,000   ├──→ Impact = 8.5/10   │   │
│  │  │ Revenue Link │ Direct   │                      │   │
│  │  └──────────────┘          │                      │   │
│  │                    ↓                              │   │
│  │          Risk Score = 0.78 × 8.5 = 6.63           │   │
│  └───────────────────────────────────────────────────┘   │
│                                                         │
│  Dynamic Risk Prioritization                            │
│  ┌───────────────────────────────────────────────────┐   │
│  │ Risk Matrix:                                      │   │
│  │   Impact                                          │   │
│  │    High │ Med  │ Med  │ High │ Critical           │   │
│  │         │ Low  │ Med  │ High │ High               │   │
│  │    Med  │ Low  │ Low  │ Med  │ High               │   │
│  │         │ Low  │ Low  │ Med  │ Med                │   │
│  │    Low  │ Neg  │ Low  │ Low  │ Med                │   │
│  │         └──────┴──────┴──────┴──────              │   │
│  │           0.2  0.4  0.6  0.8  Probability         │   │
│  │                                                   │   │
│  │ Current Issues: ● Resource A (Critical)           │   │
│  │                 ● Resource B (High)               │   │
│  │                 ○ Resource C (Medium)             │   │
│  └───────────────────────────────────────────────────┘   │
│                                                         │
│  Automated Remediation Generation                       │
│ ┌─────────────────────────────────────────────────────────────┐     │
│ │ Case-Based Reasoning System                        │     │
│ │                                      │     │
│ │ Current Case:                              │     │
│ │ • Type: Storage exposure                        │     │
│ │ • Context: Production, PII data                   │     │
│ │ • Constraints: No downtime allowed                 │     │
│ │                                      │     │
│ │ Historical Case Matching:                        │     │
│ │  ┌───────────────────────────────────────┐        │     │
│ │  │ Case DB: 10,000+ resolved issues  │        │     │
│ │  │                  │        │     │
│ │  │ Similar Cases Found: 47       │        │     │
│ │  │ Similarity > 0.85: 12         │        │     │
│ │  │ Success Rate > 90%: 8         │        │     │
│ │  └───────────────────────────────────────┘        │     │
│ │                                      │     │
│ │ Selected Remediation Strategy:                    │     │
│ │ 1. Enable Azure Private Endpoints                 │     │
│ │ 2. Restrict network access to VNet                │     │
│ │ 3. Enable audit logging                          │     │
│ │ 4. Apply encryption at rest                      │     │
│ │                                      │     │
│ │ Confidence: 94% | Est. Time: 45min | Risk: Low        │     │
│ └─────────────────────────────────────────────────────────────┘     │
│                                              │
│ Natural Language Remediation Instructions                    │
│ ┌─────────────────────────────────────────────────────────────┐     │
│ │ Generated Instructions:                          │     │
│ │                                      │     │
│ │ "To remediate the storage exposure risk:             │     │
│ │                                      │     │
│ │ 1. First, create a backup of the storage account       │     │
│ │    configuration using: az storage account show...    │     │
│ │                                      │     │
│ │ 2. Enable private endpoints by navigating to the       │     │
│ │    Azure Portal > Storage Account > Networking >        │     │
│ │    Private Endpoints. Click 'Add' and select your      │     │
│ │    VNet and subnet.                            │     │
│ │                                      │     │
│ │ 3. After the endpoint is created, disable public        │     │
│ │    access: Set 'Public network access' to 'Disabled'    │     │
│ │                                      │     │
│ │ 4. Test access from within the VNet to ensure          │     │
│ │    connectivity before proceeding.              │      │
│ │                                         │      │
│ │ Expected impact: No downtime if executed correctly.    │      │
│ │ Rollback: Re-enable public access if issues arise."   │      │
│ └──────────────────────────────────────────────────────────────┘      │
│                                              │
└────────────────────────────────────────────────────────────────────────────┘
```

Abstract

A machine learning system for predicting cloud policy compliance violations through advanced temporal analysis and configuration drift detection. The invention employs ensemble machine learning combining gradient boosting, LSTM networks with attention mechanisms, and Prophet models to achieve 90% precision in 24-hour violation predictions. A sophisticated configuration drift detection system uses variational autoencoders to model baseline configurations and identify movements toward non-compliant states through statistical process control. The temporal pattern analysis engine implements wavelet transforms, motif discovery, and causal inference to identify complex compliance patterns including seasonal variations and regime changes. Risk assessment combines calibrated probability estimates with business impact analysis using fuzzy logic aggregation, generating prioritized risk scores for thousands of cloud resources. The automated remediation engine employs case-based reasoning to match predicted violations with proven remediation strategies, generating natural language instructions with confidence scores and implementation timelines. The system processes over 100,000 compliance events per minute, enabling proactive governance management that prevents violations before occurrence rather than reacting after detection.