Enhancing NJCCIC Threat Detection: A Machine Learning Approach to Anomalous Login Behavior

Problem Statement and Goals

The New Jersey Cybersecurity and Communications Integration Cell leads and coordinates the cybersecurity efforts of New Jersey to protect all its citizens. Up to 80 percent of all breaches, including unauthorized access and human operated ransomware attacks, are the results of stolen or compromised login credentials[1]. The NJCCIC seeks to improve their current threat detection methodologies. I assisted their efforts by experimenting with machine learning capabilities for anomalous login behavior.

Goals:

- Leverage data engineering techniques for feature preparation.
- Apply Machine Learning for classification of anomalous logins.

Rosemary James

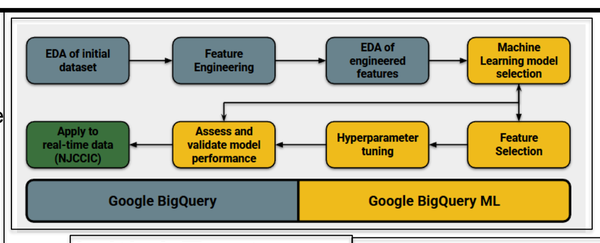
Rosemary James is a hard-working New Jersey state employee and mother of three children. Her family has access to many smart devices; television, phones, tablets, Google Home, and watches. Rosemary is always on the move trying to keep up with work and her kids. She often accesses her state account from different devices and locations daily. Sometimes she forgets to log out of her account when she is not working. Her kids love to watch educational videos on her iPhone. If Rosemary is not careful, her kids may click a phishing link and have her login credentials breached.

	Awareness	Consideration	Decision	Retention
Touchpoints	• User logs onto network as usual.	• Account password changed. • Someone hacked the account and obtained her login credentials.	• Following tips on account security given by the customer service representative of the website.	• Going to NJCCIC for tips on best practices for personal security online.
Actions	• Leaves her tablet/phone logged-in in reach of her children. • Ignoring google warning of password security.	• Consider asking customer service for help in reinstating the account. • Implications of critical changes in the account/ lost critical information. • Consult NJCCIC for reporting stolen login.	• Changing and adding passwords. • Instructing children about internet safety.	• Add MFA (Multi-factor authentication). • Not using the same passwords for everything.
Experience	Interested			
Solution	Continues as usual.	Contact costumer service representatives.	Being vigilant.	Advocate for best cybersecurity practices for everyone.

Approach

General Approach:

The NJCCIC team provided a subset of raw log files. Feature engineering techniques, such as aggregation and binning, were applied to create new features aimed at catching behavior patterns of users. There were three sets of features including our initial set, our larger 4/4a set, and our minimized 4b set. A correlation matrix was used to remove initial features that correlated too strongly with each other to avoid multicollinearity. Feature selection and hyperparameter tuning was performed when training our machine learning models to identify the best performing model.



The Autoencoder Model:

The project leveraged BigQueryML's native Autoencoder ML model specialized for anomaly detection. An autoencoder is a type of artificial neural network that learns to represent data in a compressed form and then reconstructs it as closely as possible to the original input[2].

Autoencoders consists of two components:

Encoder: This compresses the input into a compact representation and capture the most relevant information.

Decoder: It reconstructs the input data from this compressed form to make it as similar as possible to the original input.

The MSE values are a calculation of the difference between the input and the output based on similarity.

Initial Features:

waf_timestamp	datetime64[us, UTC]
principal_ip	object
user_id	object
forgerock_action	int64
principal_url	object
referral_url	object
user_browser	object
user_agent	object
carrier_name	object
longitude	float64
latitude	float64
application	object
AccountCreatedDate	object
AllUserRoles	object
EnabledMFA	boolean
RegisteredMFA	boolean
prev_EnabledMFA	boolean
MFA_disabled_recently	boolean
susp_user_browser	bool
unique_devices	int64
failed_logins	int64
login_hour	float64
login_day_name	object
account_age	int64
MFA_enabled_and_registered	boolean
1hr_login_freq	int64
time_since_last_login	float64
frequent_login_30min	int64
frequent_login_1hr	int64
frequent_login_3hr	int64
logins_past_1hr	int64
logins_past_3hr	int64
logins_till_next	float64
distinct_ip_count_per_user_id	int64
geolocation_login_frequency_per_user_id	float64
distinct_geolocation_count_per_user_id	int64
day_of_week	int32
hour_of_day	int32
geohash	object
geolocation_distinct_users_count	int64
historical_access_anomaly	bool
avg_login_hour	float64
login_hour_deviation	float64
ua_user_freq	int64
ua_user_freq_norm	float64
carrier_user_freq	int64
carrier_user_freq_norm	float64
running_unique_urls	int64

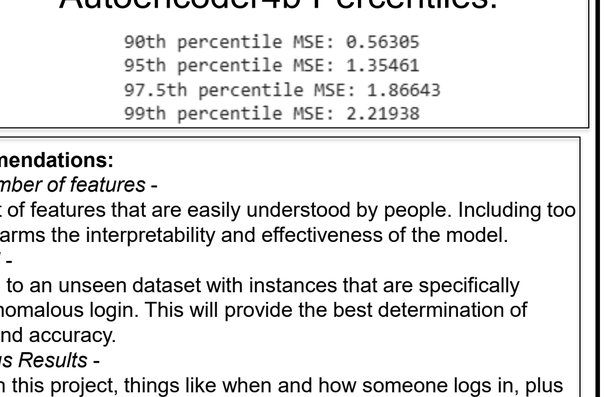
The initial feature set was used to derive new feature sets. Autoencoder 4/4a used the full set of features while Autoencoder 4b used a greatly reduced set of features.

Autoencoder4/4a Features

Field name	Type	Mode	Description				
EnabledMFA_bin	INT64	NULLABLE		geohash_drenr	INT64	NULLABLE	
MFA_disabled_recently_bin	INT64	NULLABLE		geolocation_distinct_users_count_z	FLOAT64	NULLABLE	
MFA_enabled_and_registered_bin	INT64	NULLABLE		historical_access_anomaly_bin	INT64	NULLABLE	
RegisteredMFA_bin	INT64	NULLABLE		hr_login_freq_z	FLOAT64	NULLABLE	
account_age_z	FLOAT64	NULLABLE		login_hour_deviation_z	FLOAT64	NULLABLE	
avg_login_hour_z	FLOAT64	NULLABLE		login_hour_z	FLOAT64	NULLABLE	
carrier_user_freq_norm	FLOAT64	NULLABLE		logins_past_1hr_z	FLOAT64	NULLABLE	
day_of_week_z	FLOAT64	NULLABLE		logins_past_3hr_z	FLOAT64	NULLABLE	
distinct_geolocation_count_per_user_id_z	FLOAT64	NULLABLE		logins_till_next_z	FLOAT64	NULLABLE	
distinct_ip_count_per_user_id_z	FLOAT64	NULLABLE		prev_EnabledMFA_bin	INT64	NULLABLE	
failed_logins_z	FLOAT64	NULLABLE		running_unique_urls_z	FLOAT64	NULLABLE	
forgerock_action	INT64	NULLABLE		susp_user_browser_bin	INT64	NULLABLE	
frequent_login_30min_z	FLOAT64	NULLABLE		time_since_last_login_z	FLOAT64	NULLABLE	
geohash_9qe14	INT64	NULLABLE		ua_user_freq_norm	FLOAT64	NULLABLE	
geohash_9v8vy	INT64	NULLABLE		unique_devices_z	FLOAT64	NULLABLE	
geohash_9wuhx	INT64	NULLABLE					
geohash_Other	INT64	NULLABLE					
geohash_dhyj3	INT64	NULLABLE					
geohash_dnng3	INT64	NULLABLE					
geohash_dq8hf	INT64	NULLABLE					
geohash_dqcqp	INT64	NULLABLE					
geohash_dr347	INT64	NULLABLE					
geohash_dr55f	INT64	NULLABLE					

Autoencoder4b
Features circled above

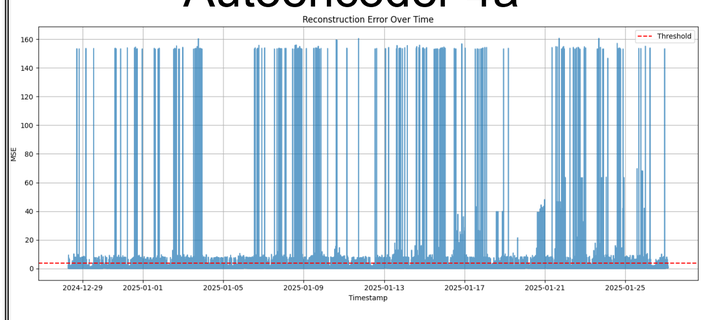
Initial Training Results



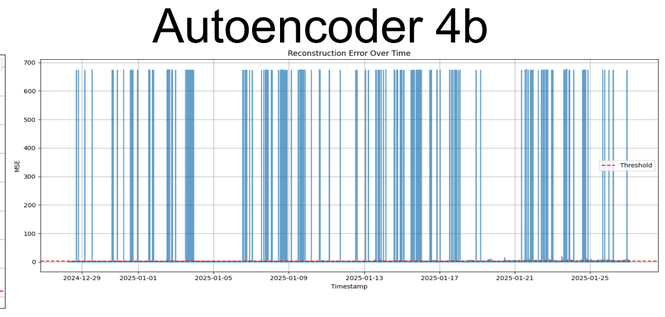
Results

The Autoencoder model operates through the DETECT_ANOMALIES method within Google's BigQueryML platform which calculates the MSE values for each login event in the dataset. A contamination value is needed to determine what is anomalous. The contamination value was determined by examining the percentile distribution of the MSE values so that 99% of the data was below the threshold. The model works best when it only marks the exceptionally high MSE values. The MSE values and contamination threshold were visualized over time to determine which model performed better. It was determined that model 4b, run on the 4b feature set, performed the best due to the lack of noise at the threshold. The charts below were derived from the results of model 4b.

Autoencoder 4a



Autoencoder 4b

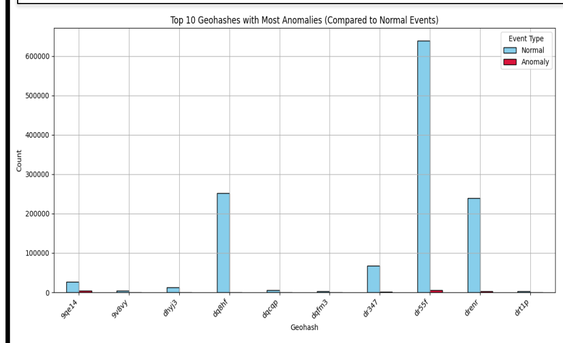


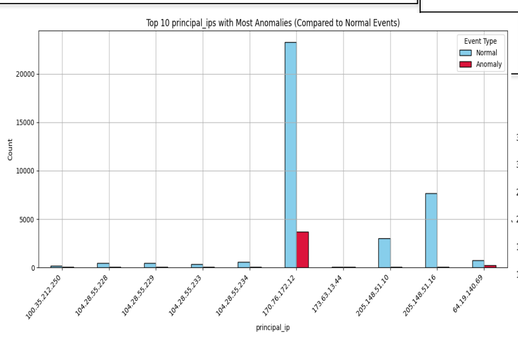
Autoencoder4a Percentiles

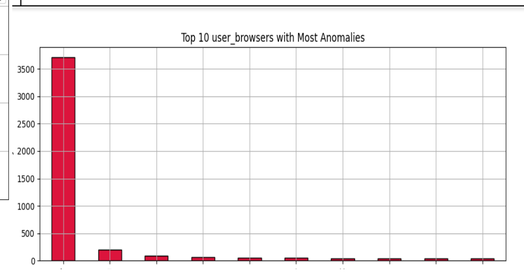
90th percentile MSE: 0.77643
95th percentile MSE: 1.65557
97.5th percentile MSE: 2.55098
99th percentile MSE: 3.81254

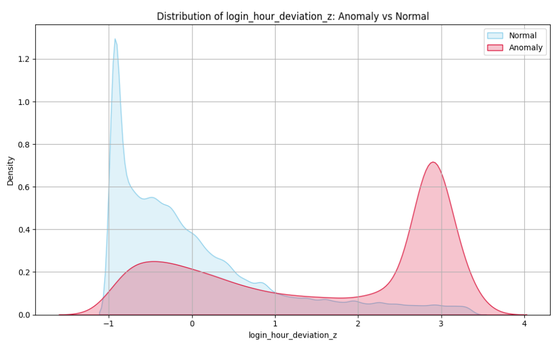
Autoencoder4b Percentiles:

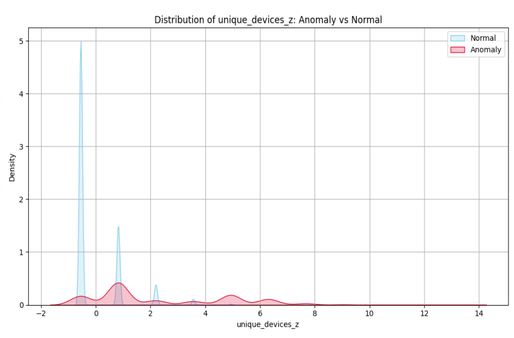
90th percentile MSE: 0.56305
95th percentile MSE: 1.35461
97.5th percentile MSE: 1.86643
99th percentile MSE: 2.21938











Note: Chart labels removed for confidentiality

Future Recommendations:

Use a select number of features -
Identify a subset of features that are easily understood by people. Including too many features harms the interpretability and effectiveness of the model.

Verify the Model -
Apply our model to an unseen dataset with instances that are specifically marked as an anomalous login. This will provide the best determination of model efficacy and accuracy.

Verify Anomalous Results -
From our work in this project, things like when and how someone logs in, plus whether they use multi-factor authentication, help us spot unusual login activity.

References:

1. New Jersey Cybersecurity & Communications Integration Cell. (n.d.). New Jersey Cybersecurity & Communications Integration Cell. Retrieved from https://www.cyber.nj.gov/
2. Gupta, A. (2025, December 23). Autoencoders in Machine Learning. GeeksforGeeks. Retrieved from https://www.geeksforgeeks.org/machine-learning/auto-encoders/

Acknowledgements:
This project was conducted through the Rutgers MBS Externship Exchange in collaboration with the New Jersey Cybersecurity & Communications Integration Cell. Special thanks to Dr. Karen Bemis, Abbe Rosenthal, Rob Bruder, Aditi Shah, Aditya Patil, Indira Rivera, Harin Ponna, and Vatsal Malkari.

Logan Buddenbaum
Rutgers University

Enhancing NJCCIC Threat Detection: A Machine Learning Approach to Anomalous Login Behavior

Abstract

State-level cybersecurity operations centers (SOCs) face increasing volumes of authentication data, making manual review of potential compromises impossible. This project, conducted in partnership with the New Jersey Cybersecurity & Communications Integration Cell (NJCCIC), developed a machine learning-based approach to detect anomalous login behaviors. By leveraging Google Cloud Platform for data engineering and applying anomaly detection algorithms to custom feature sets, including MFA status and geolocation, the team successfully identified potential malicious activity patterns that static rules and the human eye often miss. The results demonstrate that behavioral analytics provides a higher fidelity signal for threat detection, allowing SOC analysts to prioritize high-risk events more effectively.

Logan Buddenbaum
Rutgers University

RUTGERS