International Journal of Computer Applications Technology and Research
Volume 15–Issue 10, 20 – 32, 2026, ISSN: -2319–8656
DOI:10.7753/IJCATR1510.1002

A Comparative Evaluation of Classical, Hybrid, and Generative Data Augmentation Techniques for Imbalanced SQL Injection Detection

David Njuguna
Murang'a University of Technology
Murang'a, Kenya

Gabriel Kamau
Murang'a University of Technology
Murang'a, Kenya

Stephen Kahara
Murang'a University of Technology
Murang'a, Kenya

Abstract: Class imbalance remains a major challenge in developing machine learning models for SQL injection attack detection because harmful queries are usually underrepresented in real-world datasets. This study comparatively evaluates the effectiveness of classical, hybrid, and generative data augmentation techniques for improving SQL injection attack detection performance on imbalanced data. Seven augmentation methods—SMOTE, ADASYN, Borderline-SMOTE, SMOTE-Tomek Links, SMOTE-ENN, Generative Adversarial Networks, and conditional GAN- were applied to the SQLiV3 dataset and evaluated using Decision Tree, Random Forest, Naive Bayes, Support Vector Machine, and DistilBERT classifiers. Model performance was assessed using Accuracy, Precision, Recall, F1-score, Matthews correlation coefficient, and ROC-AUC. Results demonstrate that data augmentation significantly improves SQL detection compared with the original imbalanced dataset, although the degree of improvement depends on both the augmentation technique and classifiers. Among the conventional oversampling methods, SMOTE outperformed ADASYN, with Random Forest achieving a ROC-AUC of 98.16%. Among the hybrid techniques, SMOTE-ENN produced the best performance for conventional machine learning models, with Random Forest attaining 99.87% accuracy and a ROC-AUC of 99.67%. Generative Adversarial Network-based augmentation also achieved highly competitive results, with Random Forest recording 99.49% accuracy and a 99.82% ROC-AUC. Across all experiments, DistilBERT consistently outperformed conventional classifiers, achieving up to 99.87% accuracy and a ROC-AUC of 100% on the SMOTE-balanced dataset. Friedman's statistical test confirmed statistically significant differences among the augmentation techniques. These findings demonstrate that augmentation methods that improve class balance and preserve discriminative characteristics of SQL injection queries produce superior detection performance.

Keywords: SQL injection attacks, class imbalance, data augmentation, cybersecurity, SMOTE, GAN, cGAN, DistilBERT, Random Forest, cybersecurity, machine learning

1. INTRODUCTION

Structured Query Language Injection (SQLi) attacks remain among the most prevalent and destructive web application attacks, enabling attackers to manipulate database queries via malicious user input, thereby compromising the integrity, confidentiality, and availability of information systems [1]. Despite significant advances in secure software development and web application security, SQL injection continues to pose a major cybersecurity threat due to its simplicity, exploitability, and severe consequences. The OWASP Top 10 – 2023 ranks injection attacks, including SQLi, among the most critical web application security risks due to their exploitability, detectability, and impact [2], [3]. Furthermore, SQLi accounted for over 33% of reported web application breaches in 2023, resulting in substantial financial and reputational damage for the affected organizations [4].

Beyond direct database compromise, SQLi undermines the effectiveness of operational cybersecurity systems, such as Web Application Firewalls (WAFs), Intrusion Detection Systems (IDS), secure coding validation tools, and Security Operations Centers (SOCs). Sophisticated and obfuscated SQL injection payloads frequently evade signature-based mechanisms, while anomaly detection systems may generate excessive false alarms or fail to identify previously unseen attack patterns. Consequently, improving the robustness, reliability, and generalization capability of SQL injection detection models remains an important research challenge.

Machine learning (ML) and deep learning techniques have significantly improved SQL injection detection by learning complex attack patterns from historical data. However, existing detection systems struggle to effectively identify and classify polymorphic SQLi patterns in real-world settings [5], [6]. In practice, SQL injection datasets are highly imbalanced, with more benign SQL queries than malicious ones [7]. Similar class imbalance problems are also observed in other cybersecurity and real-world applications, including fraud detection, medical diagnosis, and network intrusion detection, where minority-class instances represent the majority of samples, leading to extremely skewed datasets [8]. This class imbalance skews model training, often resulting in reduced sensitivity to rare yet critical attack instances.

To address the challenge, data augmentation techniques have been proposed to improve minority-class representation... The Synthetic Minority Over-Sampling Technique (SMOTE) [9] remains one of the most widely used approaches because it creates new samples through interpolation between minority class instances. Several variants such as ADASYN [10] and Borderline-SMOTE [11] further improve synthetic sample generation by focusing on difficult-to-learn samples and decision boundary regions. While these methods improve class balance, they may also introduce noise or increase class overlap [8].

Hybrid data augmentation techniques have been proposed to overcome these limitations. Approaches such as SMOTE-

ENN [12], [13] and SMOTE combined with Tomek Links [14] integrate oversampling with data-cleaning to remove ambiguous and noisy samples, thereby enhancing class separability. More recently, deep learning-based generative models, particularly Generative Adversarial Networks (GANs), have gained attention for their ability to learn complex data distributions and generate highly realistic synthetic samples capable of improving model generalization [15], [16].

Although previous studies have demonstrated the effectiveness of individual augmentation techniques, most studies have concentrated on particular application domains, such as fault diagnosis and medicine, or on particular strategies, like cost-sensitive learning [17]- [19], rather than SQL detection. Existing studies typically evaluate only one or a few augmentation methods, focus primarily on conventional machine learning algorithms, or overlook transformer-based models that have shown good performance in cybersecurity applications. Moreover, limited research has systematically compared classical oversampling methods, hybrid resampling techniques, and deep generative augmentation approaches within a unified experimental framework using identical datasets, classifiers, and evaluation metrics. Consequently, there remains insufficient empirical evidence to guide researchers and practitioners in selecting an appropriate augmentation strategy for improving SQL injection detection performance.

To bridge this gap, this study presents a comprehensive comparative evaluation of seven widely used data augmentation techniques: SMOTE, ADASYN, Borderline-SMOTE, SMOTE-ENN, SMOTE combined with Tomek Links, GAN, and conditional GAN (cGAN). Their effectiveness is evaluated using an imbalanced SQL injection dataset and five classification models, namely Decision Tree (DT), Random Forest (RF), Naive Bayes (NB), Support Vector Machine (SVM), and DistilBERT. Performance is assessed using multiple evaluation metrics, including Accuracy, Precision, Recall, F1-score, Matthews Correlation Coefficient (MCC), and ROC-AUC, together with statistical significance testing.

This work makes three main contributions: (i) A comprehensive comparison of deep learning-based, hybrid, and classical data augmentation techniques for handling class imbalance in SQL injection datasets; (ii) An empirical assessment of the impact of these augmentation techniques on both conventional machine learning classifiers and the transformer-based DistilBERT model; and (iii) Practical recommendations for selecting appropriate augmentation strategies to improve SQL injection detection in real-world cybersecurity applications. The rest of the paper is organized as follows: Section 2 reviews related studies, Section 3 describes the research methodology, and Section 4 presents results. Section 5 discusses the results. Finally, Section 6 concludes the study and outlines recommendations.

2. LITERATURE REVIEW

2.1 Augmentation Methods for Imbalanced Learning

Class imbalance remains one of the major challenges in ML because a classifier trained on skewed datasets tends to favor the majority class while displaying poor sensitivity to minority-class instances. This problem is particularly critical in Cybersecurity applications where malicious samples are considerably fewer than benign samples. To address this limitation, numerous data augmentation techniques have been proposed to improve minority-class representation and enhance classifier performance.

2.1.1 SMOTE

SMOTE is one of the most widely adopted oversampling techniques for handling imbalanced datasets. Instead of duplicating minority-class samples, it generates synthetic samples through linear interpolation between existing minority samples and their K-nearest neighbors [20], [21]. Despite its effectiveness, SMOTE has several limitations. Because synthetic samples are generated through interpolation, the technique may introduce noisy observations, increase overlap, and generate unrealistic samples when applied to highly complex or high-dimensional datasets [8], [22]. These limitations are particularly relevant for structured SQL queries, where semantic relationships cannot always be preserved through numerical interpolation.

2.1.2. Borderline-Smote

BL-SMOTE aims to resolve the problem of generating synthetic samples in safe regions by identifying and oversampling only the "borderline" minority instances that are close to the decision boundary [23]. By concentrating synthesis on the boundary, BL-SMOTE more effectively refines the classifier's decision region, often leading to better performance metrics than standard SMOTE. However, the technique may also amplify noisy or incorrectly labelled boundary samples. Furthermore, because SQL queries are sparse, structured, and semantically constrained, distance-based interpolation may generate invalid or unrealistic SQL samples that do not accurately represent the attack behavior.

2.1.3. ADASYN

ADASYN automatically adapts the number of synthetic samples created for each minority instance based on how difficult it is for the classifier to learn that instance [10] [28], i.e., Minority samples near the class boundary or surrounded by majority samples get more synthetic samples, while minority samples in the instances perceived to be safe get fewer or no synthetic samples. ADASYN is better at handling highly imbalanced datasets with complex distributions as it adaptively shifts the decision boundary to focus on difficult samples. However, ADASYN may unintentionally

oversample noisy minority instances, resulting in increased class overlap and reduced model robustness. [22]. For SQL injection datasets, where meaningful semantic structures are more important than geometric boundaries, this behavior may produce unrealistic attack samples and degrade classifier performance.

2.1.4. SMOTE+Tomek Link

SMOTE+TomekLink combines minority-class oversampling with data cleaning. After SMOTE generates synthetic samples, Tomek Links identify neighboring samples belonging to different classes and remove ambiguous majority-class samples [24]-[27]. This process reduces class overlap and improves class separability while preserving balanced class distributions. Compared with conventional SMOTE, SMOTE-Tomek produces cleaner decision boundaries and reduces the likelihood of overfitting caused by noisy observations. Consequently, it has become one of the most widely adopted hybrid resampling techniques for imbalanced classification.

2.1.5. SMOTE-ENN

SMOTE-ENN integrates SMOTE with the Edited Nearest Neighbors (ENN) undersampling algorithm. Following oversampling, ENN removes observations whose class labels differ from the majority of their nearest neighbors [23]. Unlike Tomek Links, ENN removes noisy samples from both majority and minority classes, producing a cleaner and more discriminative training dataset [23]. Previous studies have consistently shown SMOTE-ENN to be highly effective as it both balances the data and improves data quality by reducing noise and class overlap near the boundaries. It often outperforms SMOTE alone in terms of stability, accuracy, and generalization capabilities. However, the aggressive removal of majority-class samples may substantially reduce dataset size, potentially discarding informative observations in some applications.

2.1.6. Generative Adversarial Networks (GAN) for data augmentation

GANs provide an alternative to interpolation-based oversampling by learning the underlying probability distribution of the training data [29]. A GAN consists of two neural networks: the generator that creates synthetic samples and a discriminator that distinguishes between real and generated samples [30], [31]. Through adversarial training, the generator progressively learns to produce increasingly realistic synthetic data.

Several variants of GANs have been developed to improve synthetic data generation. These include the Conditional GAN (cGAN), which works like a GAN but incorporates additional information such as class labels into the input to guide the generator's output [32], [33]. Other variants include Outlier Detectable GAN (OD-GAN) [34]. The Single-Objective Generative Adversarial Active Learning (SO-GAAL) [35], conditional Variational Autoencoder (VAE [36], and Autoencoder-Conditional GAN (AE-CGAN) [37]. Supervised Adversarial Variational Autoencoder with Regularization and Deep Neural Network (SAVAER-DNN) [38] and Wasserstein GAN (WGAN) [39]. Although. these models often generate more realistic synthetic samples than interpolation-based approaches, they require greater computational resources, longer training times, and careful hyperparameter optimization. Additionally, syntactic correctness does not always guarantee semantic validity, particularly for structured SQL queries. [40].

2.2 Data Augmentation for SQL Injection Detection

Machine learning has become an important approach for SQL injection detection because of its ability to learn complex attack patterns from historical data. Several studies have incorporated data augmentation techniques to mitigate class imbalance and improve classifier performance.

Early studies primarily employed SMOTE to balance SQL injection datasets before training conventional machine learning classifiers. For example, [41] evaluated several machine learning algorithms using SMOTE-balanced data and selected the best-performing classifier based on accuracy. Similarly, [42] demonstrated that SMOTE slightly improved Naïve Bayes performance on imbalanced SQL injection datasets. Although encouraging, both studies relied primarily on accuracy and therefore provided limited insights into classifier behavior under class imbalance.

Subsequent research integrated augmentation with hybrid detection approaches. For instance, [43] combined SMOTE with machine learning classifiers in a two-layer firewall architecture, whereas [44] evaluated SMOTE together with deep learning models such as Bi-LSTM. Although these studies reported improved detection accuracy, they did not comprehensively investigate the influence of different augmentation techniques on transformer-based architectures.

Hybrid augmentation techniques have also attracted considerable attention. The study reported in [45] employed SMOTE-Tomek Links together with feature selection and LightGBM to achieve high SQL injection performance. Similarly, [46] combined SMOTE with rule-based and machine learning approaches. While these studies demonstrated the benefits of hybrid resampling, they only evaluated a single augmentation method.

Recent studies have explored more advanced deep learning architectures. Graph neural networks combined with SMOTE and Isolation Forest were proposed in [47] to improve structural representation of SQL queries, while ensemble approaches integrating Random Forest and XGBoost were presented in [48]. Likewise, [49] demonstrated the effectiveness of ensemble learning on SMOTE-balanced datasets; although these approaches achieved high detection

accuracy, they remained dependent on conventional oversampling methods.

Generative augmentation has emerged as a promising alternative for SQL injection detection. GAN-based methods have been used to generate realistic SQL injection payloads for improving classifier performance [50]. Recently, a study by [51] combined Variational Autoencoders. Conditional Wasserstein GAN with Gradient Penalty (CWGAN-GP) and Unet to generate synthetic queries that improved detection performance while reducing false positives and false negatives.

Despite these developments, several research gaps exist. First, most existing studies evaluate only one or two augmentation techniques, rather than systematically comparing classical oversampling, hybrid resampling, and deep learning generative methods within an integrated framework.

Second, most studies focused on conventional ML algorithms, with little evaluation of transformer-based models such as DistilBERT. Lastly, fewer studies simultaneously examine how augmentation techniques influence both traditional machine learning classifiers and current language models using identical datasets, preprocessing procedures, and evaluation metrics. Addressing these gaps forms the basis for this study.

Table 1: Comparison of SMOTE Variants and GANs for Data Augmentation

Technique	Mechanism	Strengths	Limitations	Relevance to SQLi
SMOTE	Linear interpolation between minority samples	Simple, widely adopted, improves class balance	May introduce noise and class overlap	Effective baseline oversampling technique
Borderline-SMOTE	Oversampling on minority samples near the decision boundary	Improves minority class discrimination	Can still inflate dataset size, potential for noise sensitivity	Enhanced numerical balance, particularly for ambiguous SQLi samples
ADASYN	Adaptive oversampling based on learning difficulty	Focuses on difficult minority samples	May amplify noisy observations	Contributes to numerical balance, potentially more adaptive to SQLi patterns
SMOTE-Tomek	SMOTE followed by Tomek Link cleaning	Reduces class overlap and improves decision boundaries	May remove informative majority samples	Effective hybrid balancing approach
SMOTE-ENN	SMOTE followed by Edited Nearest Neighbor	Balances classes while removing noisy observations	Can aggressively reduce majority-class data	Suitable for noisy cybersecurity datasets
GAN	Learns data distribution through adversarial training	Generates diverse and realistic data, learns complex patterns	Computationally expensive, training instability	Suitable for generating realistic SQL attack samples
cGAN	Learns data distribution through adversarial training and incorporates additional information such as class labels into the input to guide the generator's output	Generates class-conditioned, diverse, and realistic data, preserving class traits	Computationally intensive, training instability	Suitable for augmenting datasets containing polymorphic and evolving SQL injection patterns

3.0 Methodology and Research Design

3.1 Research Design

The study adopted an experimental research design to analyze the effectiveness of classical, hybrid, and deep learning-based data augmentation techniques for mitigating class imbalance in SQL injection attack (SQLIA) detection. The steps involved include: selection of the dataset, dataset preprocessing, analysis of data augmentation techniques, data splitting, training ML models, model evaluation, and statistical analysis, as shown in Figure 1.

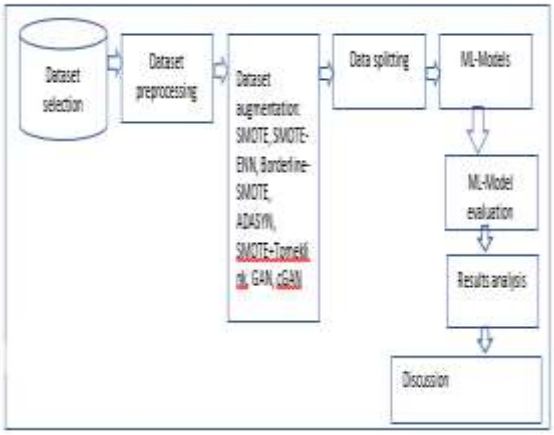

Figure 1: Augmentation analysis steps

3.2 Data Description

The dataset was obtained from Kaggle.com (SQLiV3.csv) [52]. This dataset was selected because it contains imbalanced benign and malicious SQL queries. It bears two attributes: "Query-SQL statement" and "Labels-class" (0-benign and 1-malicious). The number of malicious queries is 11382, while the benign queries are 19537, as shown in Table 2. The dataset was selected randomly

Table 2: Original SQLiV3 Dataset composition

Class	Number of Queries
	Benign queries
Benign	19537
malicious	11382

3.3 Data Pre-processing:

Raw data often contains significant amounts of noise and inconsistency that can impact model performance [37]. Therefore, the researcher performed data preprocessing, which includes cleaning the data from irrelevant entries, handling missing values, and normalizing the data to ensure it is on a consistent scale. This process ensures that the data is ready for further analysis. Besides data cleaning, tokenization and vectorization of the dataset were also performed. Feature extraction for traditional machine learning models was performed using Term Frequency–Inverse Document Frequency (TF-IDF) vectorization, which changed SQL queries into high-dimensional numerical feature vectors appropriate for classification. The pretrained DistilBERT tokenizer, which automatically handles padding, truncation, and attention masks, was used to tokenize raw SQL queries for DistilBERT.

To avoid data leakage, the dataset was divided using stratified sampling. 80% was allocated to training and 20% to testing. Testing was done on the original imbalanced dataset. All data augmentation approaches were applied only to the training split, while the testing set remained unchanged and contained only original SQL queries

3.4 Data Augmentation Techniques

Seven augmentation techniques were evaluated to address class imbalance. This include SMOTE, SMOTE-ENN, SMOTE+TOMEKlink, Borderline-SMOTE, ADASYN, GAN, cGAN. The first five methods represent classical and hybrid resampling techniques, whereas GAN and cGAN are deep learning generative techniques capable of learning the distribution of malicious queries to generate synthetic samples. Each augmentation method was independently applied to the training dataset, generating datasets used for model training

3.5 Classification Models

Five classification algorithms were evaluated, including: Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB), Support Vector Machine (SVM), and DistilBERT. The four conventional ML classifiers were selected because they represent widely used baseline classifiers for SQL injection detection. DistilBERT was included as a transformer-based language model capable of learning contextual and semantic representation from SQL queries.

The performance of the models was evaluated and compared to determine the best classifier. For DistilBERT, transfer learning was adopted by fine-tuning the pretrained DistilBERT-uncased model for binary SQL injection classification. The maximum input sequence length was set to 128 tokens. Model optimization used the AdamW optimizer with a learning rate of 2e-5, a batch size of 16, and five training epochs. Cross-entropy was used as the optimization objective.

3.6 GAN and cGAN configuration

The GAN and cGAN models used a 100-dimensional latent noise vector as input. Both the generator and discriminator were built with fully connected dense layers and ReLU activations, with batch normalization applied in the generator for stable training. The discriminator used a ReLU activation for synthetic and real sample classification. In the cGAN, class labels were incorporated into both the generator and discriminator to produce class-specific SQL queries. Both models were trained for 2000 epochs using the Adam optimizer (learning rate=0.0002, 0.5). Model performance was monitored through generator and discriminator losses, and generated SQL queries were validated for syntax and checked for duplicates before being added to the augmented training dataset.

3.7 Experimental Environment

All experiments were conducted using Google Colab Pro with TPU v5e-1. The implementation used Hugging Face Transformers, TensorFlow, and Scikit-learn. Random seeds were fixed for Python, NumPy, TensorFlow, and data splitting processes to increase reproducibility and ensure consistent dataset splits and model initialization across experiments.

3.8 Performance Evaluation

The performance of the proposed model was evaluated using accuracy, precision, recall, F1-score, ROC AUC, sensitivity, and MCC. Since SQL injection detection involves imbalanced data, MCC and ROC-AUC were included to ensure a more reliable assessment than accuracy alone. To determine whether observed differences among augmentation techniques were statistically significant, Friedman's non-parametric test was performed. Pairwise comparisons were subsequently conducted using the Wilcoxon Signed-Rank Test.

4.0 Results

International Journal of Computer Applications Technology and Research
Volume 15–Issue 10, 20 – 32, 2026, ISSN: -2319–8656
DOI:10.7753/IJCATR1510.1002

4.1 Effect of Data Augmentation on Dataset Distribution

The original dataset comprised 30,919 SQL queries, including 11,382 malicious and 19,537 benign queries. Seven data augmentation techniques were used to correct this imbalance, each of which had a distinct impact on the class distribution.

As summarized in Table 3 and illustrated in Figure2, each augmentation technique altered the class distribution differently. SMOTE and GAN generated precisely balanced datasets containing 19,537 samples per class, while ADASYN generated a nearly balanced dataset, whereas cGAN generated additional queries, achieving a perfect balance of 20637 safe and malicious queries. Hybrid techniques modified both class balance and dataset size. SMOTE-Tomek Links preserved balanced classes, while slightly reducing the total number of samples, whereas SMOTE-ENN aggressively removed samples- a notable limitation, substantially reducing the majority-class samples. This reduction is not the result of random deletion. It arises from the ENN component of SMOTE-ENN, which removes observations identified as inconsistent with the class of their neighbouring samples. In the SQL feature space, benign and malicious queries may share common lexical and structural traits, causing some benign queries located close to class boundaries to be removed during the cleaning process.

These results show that although all augmentation techniques improved class balance, they differed considerably in their sampling behavior and their treatment of noisy and overlapping observations.

Table 3. Proportion of SQLIA dataset before and after applying augmentation techniques

	Malicious SQL	Benign SQL
Original SQL dataset	11382	19537
Augmentation technique		
SMOTE	19537	19537
SMOTE-ENN	19123	3910
SMOTE+Tomelink	19443	19443
Borderline-SMOTE	15629	15629
ADASYN	19175	19537
GAN	19537	19537
cGAN	20637	20637

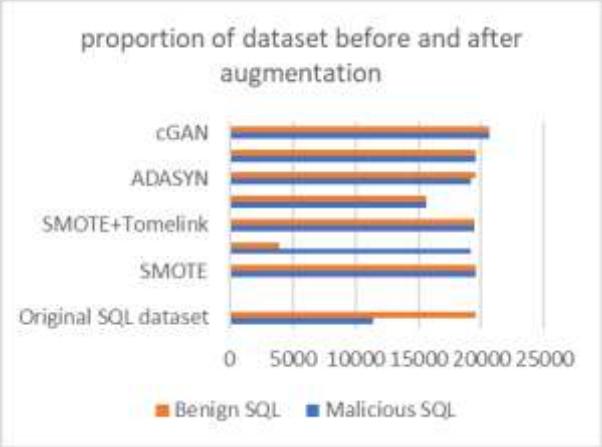

Figure 2: Visualisation of the proportion of the SQLIA dataset before and after applying augmentation techniques

4.2 Comparative Performance of ML models

Table 4 summarizes the performance of the five classifiers across the original dataset and the augmented datasets using Accuracy, Precision, Recall, F1-score, MCC, and ROC-AUC.

Table 4: Comparative Summary of ML Models Across Different Augmentation Techniques

Augmentation	Model	Accuracy	Precision	Recall/Sens	F1	MCC	ROC-AUC
Imbalanced	NB	0.9608	0.9796	0.9110	0.9440	0.9153	0.9843
	SVM	0.9388	0.9743	0.8540	0.9102	0.8684	0.9800
	DT	0.7732	0.6169	0.9911	0.7605	0.6244	0.9740
	RF	0.7708	0.6142	0.9929	0.7589	0.6223	0.9651
	DistilBert	0.999138	0.998245	0.999414	0.998829	0.998147	0.99999
SMOTE	DT	0.9519	0.9945	0.9098	0.9502	0.9072	0.9816
	RF	0.9520	0.9936	0.9108	0.9504	0.9073	0.9871
	NB	0.9497	0.9826	0.9166	0.9485	0.9016	0.9793
	SVM	0.9443	0.9872	0.9014	0.9424	0.8922	0.9696
	DistilBert	0.9991	0.9985	0.9997	0.9991	0.9982	1.0000
ADASYN	DT	0.9259	0.9870	0.8605	0.9194	0.8583	0.9724
	RF	0.9237	0.9780	0.8642	0.9176	0.8526	0.9764
	NB	0.9268	0.9819	0.8671	0.9209	0.8589	0.9722
	SVM	0.8800	0.9659	0.7838	0.8653	0.7728	0.9320
	DistilBert	0.987350	0.998694	0.976008	0.987221	0.974953	0.996710
Borderline-SMOTE	DT	0.9528	0.9540	0.9528	0.9523	0.8989	0.9779
	RF	0.9552	0.9564	0.9552	0.9548	0.9041	0.9823
	NB	0.9512	0.9532	0.9512	0.9505	0.8959	0.9750
	SVM	0.9570	0.9578	0.9570	0.9566	0.9077	0.9824

	DistilBert	0.999104	0.998466	0.999744	0.999105	0.998209	0.999979
SMOTE + Tomek	DT	0.9522	0.9922	0.9116	0.9502	0.9074	0.9828
	RF	0.9523	0.9908	0.9132	0.9504	0.9504	0.9871
	NB	0.9515	0.9840	0.9180	0.9499	0.9051	0.9806
	SVM	0.9456	0.9885	0.9019	0.9432	0.8947	0.9674
	DistilBert	0.982374	0.979213	0.972759	0.975975	0.962069	0.996195
SMOTE-ENN	DT	0.9972	0.9969	0.9997	0.9983	0.9899	0.9922
	RF	0.9987	0.9984	1.0000	0.9992	0.9954	0.9967
	NB	0.9729	0.9687	0.9997	0.9840	0.9016	0.9978
	SVM	0.9961	0.9953	1.0000	0.9977	0.9861	0.9998
	DistilBert	0.982536	0.980071	0.972319	0.976180	0.962413	0.996002
GAN	DT	0.9896	0.9940	0.9850	0.9895	0.9793	0.9939
	RF	0.9949	0.9977	0.9920	0.9948	0.9898	0.9982
	NB	0.8258	0.7470	0.9801	0.8478	0.6861	0.9883
	SVM	0.9803	0.9730	0.9876	0.9802	0.9607	0.9969
	DistilBert	0.979624	0.978203	0.966169	0.972149	0.956132	0.995299
CGAN	DT	0.942217	0.947588	0.936350	0.94192	0.884497	0.936188
	RF	0.968141	0.946457	0.992498	0.96893	0.937390	0.985685
	NB	0.812477	0.901492	0.702081	0.78939	0.640949	0.852674
	SVM	0.936887	0.942633	0.930542	0.93655	0.873847	0.974480
	DistilBert	0.980595	0.985586	0.961336	0.973309	0.958252	0.995569

4.2.1 Performance on the Original imbalanced SQL dataset

The imbalanced dataset adversely affected the performance of most conventional ML models. Among the models, Naïve Bayes achieved the best overall balance, producing the highest Accuracy, MCC, and ROC-AUC. In contrast, DT and RF exhibited extremely high Recall but comparatively low precision, indicating high positive rates.

In contrast, DistilBERT outperformed all conventional classifiers, achieving excellent performance across all metrics. These results demonstrate the ability of transformer-based language models to capture contextual and semantic information from SQL queries even under class imbalance,

4.2.2 Performance after Classical Oversampling

Classical oversampling techniques significantly improved classifier performance compared with the original dataset. SMOTE consistently produced the strongest overall performance among the conventional oversampling methods, particularly for RF, which achieved the highest ROC-AUC among the traditional machine learning models and the DistilBERT model.

Although ADASYN improved minority-class detection, its performance remained slightly lower than SMOTE across most classifiers, indicating that adaptive oversampling introduced additional variability into the generated samples. Borderline-SMOTE also produced consistent improvements, with SVM and RF demonstrating strong performance.

Across all classical augmentation methods, DistilBERT remained the highest-performing classifier, maintaining near-perfect Accuracy, Precision, Recall, MCC, and ROC-AUC values.

4.2.3 Performance after Hybrid Augmentation

Hybrid augmentation techniques produced the most consistent improvements among the conventional ML models.

SMOTE-Tomek Links improved classifier robustness by simultaneously balancing the dataset and reducing class overlap. RF achieved the strongest overall performance among the conventional classifiers under this augmentation technique.

SMOTE-ENN produced the best overall results for conventional machine learning. RF achieved the highest Accuracy, while SVM recorded the highest ROC-AUC. The aggressive removal of samples by ENN substantially improved class separability and enabled the classifiers to learn more discriminative decision boundaries. Despite the aggressive removal of majority samples, the results indicate that the retained observations contained sufficient information for highly accurate SQL injection classification. However, the high predictive performance doesn't by itself establish that the removed benign queries were redundant; therefore, aggressive reduction of benign queries is noted as a drawback of SMOTE-ENN. DistilBERT did not benefit from SMOTE-ENN to the same extent as the conventional classifiers, indicating that transformer-based models may require less aggressive resampling because of their superior contextual representation capabilities.

4.2.4 Performance after Generative Augmentation

GAN-based augmentation demonstrated highly competitive performance. RF achieved the best results among the conventional classifiers, followed by SVM, indicating that the generated SQL queries effectively enhanced the classifiers' generalization.

International Journal of Computer Applications Technology and Research
Volume 15–Issue 10, 20 – 32, 2026, ISSN: -2319–8656
DOI:10.7753/IJCATR1510.1002

Although cGAN also improved detection performance, its results were generally lower than those obtained using conventional GAN augmentation for the traditional ML models. This may be attributed to additional constraints on class-conditioned generation. Nevertheless, DistilBERT continued to achieve superior overall performance, demonstrating the robustness of transformer-based architectures regardless of the augmentation technique used. cGAN demonstrated competitive performance with DistilBERT, indicating that its generated samples retained useful semantic features

In summary, the experimental results indicate that augmentation techniques substantially improve SQL injection detection performance, although their effectiveness depends on both the augmentation technique and the classification algorithm. Among the conventional ML models, SMOTE-ENN and GAN produced the highest overall results, while DistilBERT consistently achieved the highest performance across nearly all experimental settings.

4.3. Confusion matrix analysis

Figure 3 below shows the confusion matrix for the DistilBERT model with the SMOTE-augmented SQL dataset, which produced the best overall performance among the ML models.

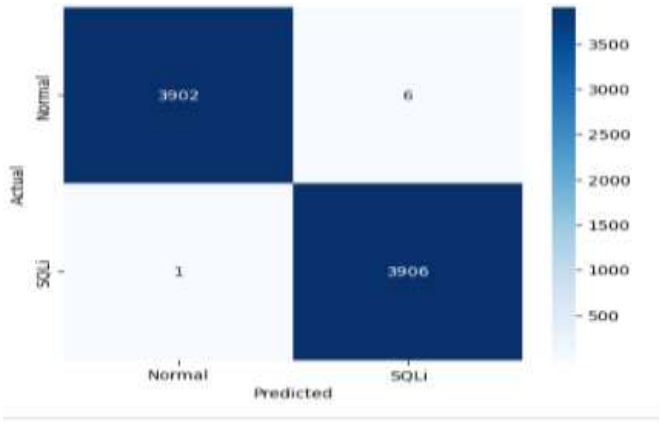

Figure 3: Confusion matrix for DistilBERT model using SMOTE dataset

The model correctly classified 3,902 normal queries and 3,906 SQL injection queries. Only six normal queries were incorrectly classified as SQL injection attacks, representing false positives, while only one SQL injection query was misclassified as normal, representing a false negative. Consequently, the model achieved an overall classification accuracy of approximately 99.91%, with SQLi precision of 99.85%, recall of 99.97%, F1-score of approximately 99.91%, and specificity of 99.85%. While the very small number of false positives suggests that the model produced very few false positives, the incredibly low number of false negatives shows that it was quite successful in identifying SQL injection attacks. These findings show that integrating DistilBERT with SMOTE-based data balancing significantly enhances the model's capacity to distinguish between benign and

4.4. Statistical Analysis

Table 5 shows the Per-class classification performance of the DistilBERT model using the SMOTE-augmented SQL dataset.

Table 5: Per-class classification performance for the DistilBERT model with SMOTE

Class	Precision	Recall	F1-score	Support	Misclassified
Benign	99.97	99.85	99.91	3908	6
Malicious	99.85	99.97	99.91	3907	1

Friedman's non-parametric test was done to determine whether the observed performance differences among augmentation techniques were statistically significant. The results indicated a statistically significant difference among the evaluated augmentation methods (χ^2 (7)=14.40, p=0.045)

Pairwise comparison using the Wilcoxon Signed-Rank test showed that SMOTE differed significantly from the imbalanced dataset (p=0.043), confirming the positive impact of class balancing on SQL injection detection. Significant differences were also observed between SMOTE and Borderline-SMOTE (p=0.042), Borderline-SMOTE and cGAN(p=0.043), GAN and SMOTE-ENN (p=0.043), and between SMOTE and cGAN (p=0.043). The absence of additional statistically significant differences (p<0.05) in all other augmentation techniques is likely due to the limited number of paired observations used in the analysis.

4.5. Descriptive Statistics

Table 6 summarizes the precision and standard deviation attained by the five classifiers for each augmentation method.

SMOTE achieved the highest average Precision, whereas SMOTE-Tomek Link produced the lowest standard deviation (SD), indicating the most consistent performance. In contrast, the original imbalanced dataset exhibited both the lowest mean Precision and the largest variation among classifiers. Although GAN and cGAN improved average Precision considerably, their SD suggest greater sensitivity to the choice of classification.

Table 6: Summary of Precision and Standard Deviations across Data Augmentation techniques

Augmentation technique	Imbalanced dataset	smote	ADASYN	BORDER-LINE SMOTE	SMOTE+TOMEK	SMOTE-ENN	GAN	cGAN
mean	0.8366	0.9916	0.98226	0.96397	0.98694	0.9879	0.9380	0.9448
SD	0.2020	0.0068	0.011959	0.019369	0.0053	0.013	0.1073	0.0298

5.0 Discussion

This study investigated the effectiveness of seven data augmentation techniques in addressing class imbalance for SQL injection attack detection using conventional machine learning classifiers and a transformer-based language model. The experimental results show that data augmentation techniques are vital for enhancing machine learning models' ability to identify SQL injection.

Class imbalance is still a significant problem in cybersecurity datasets, as evidenced by the low performance seen on the original imbalanced dataset. Conventional classifiers- Decision Tree and Random Forest in particular- showed strong recall but relatively low precision, suggesting a propensity to overpredict the minority class and produce an excessive number of false alarms. In operational settings, such as intrusion detection systems and Security Operations Centers, where high false-positive rates increase analyst burden and slow incident response, such behavior is undesirable.

Among the traditional oversampling methods, SMOTE consistently yielded the best overall performance. SMOTE enabled classifiers to learn more representative decision boundaries without significantly increasing class overlap by using interpolation to create synthetic minority-class data. As a result, Random Forest outperformed all other traditional machine learning models after SMOTE augmentation. While both ADASYN and Borderline-SMOTE enhanced classification performance, their adaptive sampling techniques seemed to introduce more variability than standard SMOTE. While Borderline-SMOTE concentrates on decision boundaries that may already contain mislabeled or overlapping instances, ADASYN concentrates synthetic sample generation around difficult-to-learn samples, which may unintentionally amplify noisy or ambiguous SQL queries. These findings are consistent with earlier research suggesting that adaptive oversampling techniques can be more susceptible to noise than conventional SMOTE [10], [22], and [23].

Hybrid augmentation techniques achieved better results for conventional ML models. Specifically, Random Forest achieved 99.87% accuracy and 99.67% ROC-AUC, whereas SMOTE-ENN yielded the best overall performance.

The additional functions of data cleansing and oversampling are responsible for the better performance. Edited Nearest Neighbors improves class separability and lowers classifier uncertainty around decision boundaries by eliminating noisy and ambiguous observations. However, a significant percentage of benign SQL queries were also eliminated by SMOTE-ENN, indicating a possible risk of over-cleaning that could limit generalization to unidentified benign samples and decrease the dataset size.

Additionally, GAN-based augmentation yielded extremely competitive outcomes. In contrast to interpolation-based oversampling methods, GANs create new synthetic samples by learning the underlying probability distribution of the minority class. More varied SQL injection patterns, including previously unknown assault variations that more accurately reflect adversarial behavior in the actual world, can be produced [53]. As a result, after GAN augmentation, Random Forest and Support Vector Machine showed outstanding classification performance. However, because unstable optimization, mode collapse, and convergence issues can lower the diversity and realism of generated samples, conventional GANs need to be trained carefully. Even though cGAN offers more control by conditioning sample production on class labels [54], its performance was relatively lower than that of the traditional GAN, which may be due to additional constraints on class-conditioned generation.

DistilBERT [55] demonstrated exceptional robustness to class imbalance by consistently outperforming Naive Bayes, SVM, Decision Tree, and Random Forest across all augmentation techniques. With near-perfect ROC-AUC values and accuracies above 99.9%, the model performed best under the imbalanced dataset, SMOTE, and Borderline-SMOTE. These results show that transformer-based contextual embeddings outperform traditional machine learning techniques in capturing intricate SQL query patterns and semantic linkages. Additionally, the findings indicate that simple oversampling techniques like SMOTE and Borderline-SMOTE complement DistilBERT more effectively than intricate generative techniques, making them appropriate options for SQL injection attack detection systems.

The statistical analysis further strengthens these findings. Friedman's test demonstrated that the observed differences among augmentation techniques were statistically significant, while the Wilcoxon Signed-Rank Test demonstrated significant improvements between the imbalanced dataset and the SMOTE-augmented dataset. Although only a limited number of pairwise comparisons reached statistical significance, this outcome is likely attributable to the relatively small number of classifiers included in the statistical analysis rather than the absence of meaningful performance differences.

Despite these encouraging results, several limitations should be acknowledged. First, the experiments were conducted using a single SQL injection dataset, which may limit the generalizability of the findings to other datasets and web application environments. Second, although the generated SQL queries improved classification performance, their semantic realism was not formally evaluated by domain experts. Third, computational efficiency, training time, and resource utilization of the augmentation techniques were not compared. Future research should therefore investigate cross-dataset validation, semantic quality assessment of generated

SQL queries, adversarial robustness against previously unseen attack variants, and explainable artificial intelligence techniques to improve transparency in SQL injection detection models.

6.0 Conclusion and recommendations

The study provided a thorough comparative analysis of conventional oversampling, hybrid resampling, and deep learning-based data augmentation strategies to solve the challenge of class imbalance in SQLi datasets. In particular, the capacity of SMOTE, ADASYN, Borderline-SMOTE, SMOTE-Tomek Links, SMOTE-ENN, GAN, and cGAN to balance the dataset and enhance the prediction performance of Decision Tree, Random Forest, Naive Bayes, Support Vector Machine, and DistilBERT classifiers was assessed.

The findings show that class imbalance significantly affects SQL injection detection performance, especially for traditional machine learning models, resulting in weaker discrimination ability, higher false-positive rates, and decreased precision. Applying data augmentation techniques showed a considerable improvement in classification performance, demonstrating that balanced datasets allow classifiers to learn more representative decision boundaries. SMOTE-ENN outperformed the other traditional augmentation techniques because it combines noise reduction and synthetic oversampling for traditional machine learning classifiers, including Random Forest and SVM. Traditional oversampling methods such as SMOTE and Borderline-SMOTE also produced consistent improvements while maintaining more balanced class distributions. GAN-based augmentation proved to be quite successful in generating diverse synthetic SQL injection samples capable of improving classifier generalization beyond simple interpolation methods.

Across all experiments, DistilBERT consistently outperformed the conventional ML classifiers, achieving the most robust and reliable performance under both imbalanced and augmented datasets. These results demonstrate that transformer-based language models are highly effective in capturing the contextual and semantic characteristics of SQL queries and can substantially improve SQL injection detection performance when combined with appropriate augmentation techniques

The study demonstrates that the ML algorithm and data augmentation strategy selection have a major impact on SQL injection detection performance. The results offer researchers and cybersecurity professionals useful advice on how to choose appropriate augmentation techniques and classification models for an imbalanced SQL injection dataset, while highlighting the growing importance of transformer-based models for web-based application security.

Based on the findings, the following recommendations are proposed:

- Researchers should evaluate augmentation techniques using multiple benchmark SQL injection datasets to assess the generalizability and robustness of the proposed approaches.

- Hybrid augmentation techniques that can achieve class balance and also preserve the syntactical and semantic structure of the dataset should be considered for ML models.

- Future studies should incorporate semantic validation of synthetic SQL queries to ensure that generated samples remain both syntactically and semantically realistic

- Transformer-based models such as DistilBERT should be prioritized for SQL injection detection due to their ability to capture contextual query patterns and their consistent performance across different augmentation techniques.

- Future research should investigate explainable artificial intelligence (XAI), adversarial robustness evaluation, and emerging generative models to develop more robust and interpretable SQL injection detection systems

ACKNOWLEDGMENTS

We sincerely thank all the people and organizations that evaluated our study paper and offered helpful criticism. Your insightful suggestions enhanced the caliber and thoroughness of our work. This study report would not have been feasible without the combined efforts and assistance of all these people and organizations

REFERENCES

[1] Sucuri, “OWASP Top 10 Security Risks & Vulnerabilities 2020,” 2020. [Online]. Available: https://sucuri.net/guides/owasp-top-10-security-vulnerabilities-2020/. [Accessed: Dec. 4, 2024].

[2] H. Ibrahim, S. Karabatak, and A. A. Abdullahi, “A study on cybersecurity challenges in e-learning and database management system,” in *Proc. 8th Int. Symp. Digital Forensics and Security (ISDFS)*, Jun. 2020, pp. 1–5.

[3] OWASP Foundation, “OWASP Desktop App Security Top 10,” 2025. [Online]. Available: https://owasp.org/www-project-desktop-app-security-top-10/. [Accessed: Jun. 28, 2025].

[4] IBM Security, Cost of a Data Breach Report 2023. Armonk, NY, USA: IBM Corporation, 2023. [Online]. Available: https://www.ibm.com/reports/data-breach. [Accessed: Jul. 15, 2026].

[5]N. Augustine, A. B. M. Sultan, M. H. Osman, and K. Y. Sharif, “Application of artificial intelligence in detecting SQL injection attacks,” *Int. J. Informatics Visualization (JOIV)*, vol. 8, no. 4, pp. 2131–2138, 2024.

[6] M. Alqhtani, D. Alghazzawi, and S. Alarifi, “Black-box adversarial attacks against SQL injection detection model,” Contemporary Mathematics, pp. 5098–5112, 2024.

[7] B. Krawczyk, “Learning from imbalanced data: Open challenges and future directions,” *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016.

[8] A. G. Udu, M. T. Salman, M. K. Ghalati, A. Lecchini-Visintini, D. R. Siddle, and H. Dong, “Emerging SMOTE and GAN-variants for data augmentation in imbalance machine learning tasks: A review,” IEEE Access, 2025.

[9] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[10] H. He, Y. Bai, E. A. Garcia, and S. Li, “ADASYN: Adaptive synthetic sampling approach for imbalanced learning,” in *Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN)*, Hong Kong, China, Jun. 2008, pp. 1322–1328.

[11] Y. Sun, H. Que, Q. Cai, J. Zhao, J. Li, Z. Kong, and S. Wang, "Borderline SMOTE algorithm and feature selection-based network anomaly detection strategy," *Energies*, vol. 15, no. 13, Art. no. 4751, 2022.

[12] M. M. Nishat, F. Faisal, I. J. Ratul, A. Al-Monsur, A. M. Ar-Rafi, S. M. Nasrullah, M. T. Reza, and M. R. H. Khan, "A comprehensive investigation of the performances of different machine learning classifiers with SMOTE-ENN oversampling technique and hyperparameter optimization for imbalanced heart failure dataset," *Scientific Programming*, vol. 2022, Art. no. 3649406, 2022.

[13] H. Hairani and D. Priyanto, "A new approach of hybrid sampling SMOTE and ENN to the accuracy of machine learning methods on unbalanced diabetes disease data," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 8, 2023.

[14] M. Gupta, K. Rajnish, and V. Bhattacharjee, “Software fault prediction with imbalanced datasets using SMOTE-Tomek sampling technique and genetic algorithm models,” Multimedia Tools and Applications, vol. 83, no. 16, pp. 47627–47648, 2024.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139-144, 2020.

[16] G. Agrawal, A. Kaur, and S. Myneni, "A review of generative models in generating synthetic attack data for cybersecurity," Electronics, vol. 13, no. 2, Art. no. 322, 2024.

[17] H. Chen, J. Wei, H. Huang, Y. Yuan, and J. Wang, "Review of imbalanced fault diagnosis technology based on generative adversarial networks," Journal of Computational Design and Engineering, vol. 11, no. 5, pp. 99-124, 2024.

[18] M. Salmi, D. Atif, D. Oliva, A. Abraham, and S. Ventura, "Handling imbalanced medical datasets: Review of a decade of research," Artificial Intelligence Review, vol. 57, no. 10, Art. no. 273, 2024.

[19] I. Araf, A. Idri, and I. Chairi, "Cost-sensitive learning for imbalanced medical data: A review," *Artificial Intelligence Review*, vol. 57, no. 4, Art. no. 1, 2024.

[20] Y. Desnelita, "Synthetic minority oversampling technique (SMOTE) for boosting the accuracy of C4.5 algorithm model," 2024.

[21] A. Muneer, R. F. Ali, A. Alghamdi, S. M. Taib, A. Almaghthawi, and E. A. Ghaleb, "Predicting customer churn in banking industry: A machine learning approach," Indonesian Journal of Electrical Engineering and Computer Science, vol. 26, no. 1, pp. 539-549, 2022.

[22] J. Brandt and E. Lanzén, "A comparative review of SMOTE and ADASYN in imbalanced data classification," 2021.

[23] M. M. Nishat, F. Faisal, I. J. Ratul, A. Al-Monsur, A. M. Ar-Rafi, S. M. Nasrullah, M. T. Reza, and M. R. H. Khan, “A comprehensive investigation of the performances of different machine learning classifiers with SMOTE-ENN oversampling technique and hyperparameter optimization for imbalanced heart failure dataset,” Scientific Programming, vol. 2022, no. 1, Art. no. 3649406, 2022.

[24] Y. Zhang, Y. Zhu, J. Liu, W. Yu and C. Jiang, "An Interpretability Optimization Method for Deep Learning Networks Based on Grad-CAM," in IEEE Internet of Things Journal, vol. 12, no. 4, pp. 3961-3970, 15 Feb.15, 2025, doi: 10.1109/JIOT.2024.3485765

[25] AIONCLOUD, “2025-03 Web Attack Trend Report,” AIONCLOUD. [Online]. Available: https://www.aioncloud.com/2025-03-web-attack-trend-report/ (accessed Nov. 27, 2025).

[26] Q. Leng, J. Guo, J. Tao, X. Meng, and C. Wang, “OBMI: Oversampling borderline minority instances by a two-stage Tomek link-finding procedure for class imbalance problem,” Complex & Intelligent Systems, vol. 10, no. 4, pp. 4775–4792, 2024.

[27] E. F. Swana, W. Doorsamy, and P. Bokoro, “Tomek link and SMOTE approaches for machine fault classification with an imbalanced dataset,” Sensors, vol. 22, no. 9, p. 3246, 2022

[28] Z. Chen, L. Zhou, and W. Yu, “ADASYN–Random forest-based intrusion detection model,” in Proc. 4th Int. Conf. Signal Processing and Machine Learning (SPML), Aug. 2021, pp. 152–159.

[29] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger and H. Greenspan, "Synthetic data augmentation using GAN for improved liver lesion classification," 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, USA, 2018, pp. 289-293, doi: 10.1109/ISBI.2018.8363576.

[30] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," IEEE Trans. Knowl. Data Eng., vol. 35, no. 4, pp. 3313–3332, Apr. 2023, doi: 10.1109/TKDE.2021.3130191.

[31] F. Omer Albasheer, R. Ramesh Haibatti, M. Agarwal and S. Yeob Nam, "A Novel IDS Based on Jaya Optimizer and Smote-ENN for Cyberattacks Detection," in IEEE Access, vol. 12, pp. 101506-101527, 2024, doi: 10.1109/ACCESS.2024.3431534.

[32] G. Douzas and F. Bacao, "Effective data generation for imbalanced learning using conditional generative adversarial networks," Expert Syst. Appl., vol. 91, pp. 464–471, 2018.

[33] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," Nov. 06, 2014, arXiv: arXiv:1411.1784. doi: 10.48550/arXiv.1411.1784

[34] J.-H. Oh, J. Y. Hong, and J.-G. Baek, "Oversampling method using outlier detectable generative adversarial network," Expert Syst. Appl., vol. 133, pp. 1–8, 2019.

[35] Y. Liu et al., "Generative adversarial active learning for unsupervised outlier detection," IEEE Trans. Knowl. Data Eng., vol. 32, no. 8, pp. 1517–1528, 2019.

[36] V. A. Fajardo et al., "On oversampling imbalanced data with deep conditional generative models," Expert Syst. Appl., vol. 169, p. 114463, 2021.

[37]J. Lee and K. Park, "AE-CGAN model based high performance network intrusion detection system," Appl. Sci., vol. 9, no. 20, p. 4221, 2019.

[38]Y. Yang, K. Zheng, B. Wu, Y. Yang, and X. Wang, "Network intrusion detection based on supervised adversarial variational auto-encoder with regularization," IEEE Access, vol. 8, pp. 42169–42184, 2020.

[39]R. Bhat and R. Nanjundegowda, "A review on comparative analysis of generative adversarial networks' architectures and applications," J. Robot. Control JRC, vol. 6, no. 1, pp. 53–64, 2025.

[40]M. Patil, M. M. Patil, and S. Agrawal, "WGAN for Data Augmentation," in GANs for Data Augmentation in Healthcare, A. Solanki and M. Naved, Eds., Cham: Springer International Publishing, 2023, pp. 223–241. doi: 10.1007/978-3-031-43205-7_13.

[41]M. H. A. AL-Maliki and M. N. Jasim, "Comparison study for NLP using machine learning techniques to detect SQL injection vulnerabilities," Int. J. Nonlinear Anal. Appl., vol. 14, no. 8, pp. 283–290, 2023.

[42] A. Arnap, "Enhancing SQL Injection Attack Detection Using Naïve Bayes and SMOTE Method on Imbalanced Datasets," J. Artif. Intell. Eng. Appl. JAIEA, vol. 4, no. 1, pp. 74–81, 2024.

[43] A. S. S. Ahmed, M. Shachi, A. A. Brishty, N. Siddiqui, and N. Sakib, "A hybrid approach to detect injection attacks on server-side applications using data mining techniques," in 2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI), IEEE, 2021, pp. 1–6.

[44] T. M. Ferreira, T. J. Lucas, C. A. C. Tojeiro, C. E. S. Bertazzoli, A. de Souza Lopes, and A. C. S. Pontara, "Detection of SQL Injection: A Comparative Analysis of Machine Learning and Deep Learning Algorithms," in Congresso Latino-Americano de Software Livre e Tecnologias Abertas (Latinoware), SBC, 2025, pp. 407–415.

[45] M. Sahu and K. P. Lilaramani, "A Review on Improved SQL Injection Detection Using Machine Jaya-Based Feature Selection and Bi-LSTM," International Journal for Multidisciplinary Research, vol. 7, no. 2, Mar.–Apr. 2025, doi: 10.36948/ijfmr.2025.v07i02.42747

[46]P. Sri-thong and C. Bunkhumpornpat, "Improving Detection of SQL Injection Attack by SMOTE," in Proc. 27th Int. Computer Science and Engineering Conference (ICSEC), 2023, doi:10.1109/ICSEC59635.2023.10329708

[47] M. K and U. P. P. S., "Enhancing SQL Query Security using Graph-based Hierarchical Embeddings with SMOTE and Advanced Neural Networks for Robust Detection," 2025 12th International Conference on Computing for Sustainable Global Development (INDIACom), Delhi, India, 2025, pp. 1-7, doi: 10.23919/INDIACom66777.2025.11115253.

[48] T. S and K. France, "SQL Injection Attack Detection in Web Applications Using Machine Learning Algorithms," 2025 8th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2025, pp. 545-552, doi: 10.1109/ICOEI65986.2025.11013708.

[49]Le, T.T.H.; Kim, H.; Kang, H.; Kim, H. Classification and explanation for intrusion detection system based on ensemble trees and SHAP method. Sensors 2022, 22, 1154.

[50]D. Lu, J. Fei, L. Liu and Z. Li, "A GAN-based Method for Generating SQL Injection Attack Samples," 2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 2022, pp. 1827-1833, doi: 10.1109/ITAIC54216.2022.9836726

[51]N. S. Dasari, A. Badii, A. Moin, and A. Ashlam, "Enhancing SQL Injection Detection and Prevention Using

Generative Models," Feb. 07, 2025, arXiv: arXiv:2502.04786. doi: 10.48550/arXiv.2502.04786.

[52] "SQL injection dataset." Accessed: Feb. 27, 2026. [Online]. Available: https://www.kaggle.com/datasets/syedsaqlainhussain/sql-injection-dataset

[53] M. Zheng et al., "Conditional Wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification," Inf. Sci., vol. 512, pp. 1009–1023, 2020.

[54] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," arXiv preprint arXiv:1910.01108, Oct. 2019

[55] S. K. Akpatsa, H. Lei, X. Li, V.-H. K. S. Obeng, E. M. Martey, P. C. Addo, and D. D. Fiawoo, "Online News Sentiment Classification Using DistilBERT," Journal of Quantum Computing, vol. 4, no. 1, pp. 1–11, 2022, doi: 10.32604/jqc.2022.026658.