

GOOSE feature engineering with optimized LSTM for DoS detection in IEC 61850 substations

Tasmi Tasmi^{1*}, Deris Stiawan², Bhakti Yudho Suprpto¹, Herri Setiawan³,
M. Agus Syamsul Arifin⁴, Rahmat Budiarto⁵, Andi Adriansyah⁶

¹Faculty of Engineering, Universitas Sriwijaya, Indonesia

²Faculty of Computer Science, Universitas Sriwijaya, Indonesia

³Faculty of Computer Science and Science, Universitas Indo Global Mandiri, Indonesia

⁴Faculty of Engineering, Universitas Jenderal Soedirman, Indonesia

⁵College of Computer Science and IT, Al-Baha University, Saudi Arabia

⁶Department of Electrical Engineering, Faculty of Engineering, Universitas Mercu Buana, Indonesia

Abstract

The IEC 61850 protocol uses GOOSE messages for fast, reliable power system communication, but is vulnerable to DoS attacks that can disrupt protection and control communications. To address this vulnerability, an LSTM model can learn sequence-based communication patterns, but its performance depends on hyperparameter configuration. Therefore, an approach is needed to optimize the LSTM configuration. The present study proposes an optimized LSTM system for detecting DoS attacks in GOOSE communications. The proposed approach integrates GOOSE-specific feature extraction, feature selection, class imbalance handling, and hyperparameter optimization to enhance the model's detection capabilities. The novelty of this research lies in the use of GOOSE communication attribute characteristics, such as stNum, sqNum, and timeAllowedToLive, within the Optimized LSTM approach to improve the reliability of DoS attack detection in GOOSE. The evaluation results demonstrate that the Optimized LSTM model exhibits a high degree of accuracy, with a mean accuracy of 99.76%, a precision of 99.84%, a recall of 99.69%, and an F1-score of 99.76%. These findings are corroborated by statistical analysis, which reveals significant disparities between models in terms of accuracy, F1-score, and ROC-AUC (p -value < 0.0001). Moreover, the Optimized LSTM model has been demonstrated to surpass competing models, including the Convolutional Neural Network (CNN), Gated Recurrent Unit (GRU), and Transformer models (adjusted p -value 0.0059–0.0176). These results support the conclusion that optimizing the LSTM model's hyperparameters can enhance DoS detection performance and advance deep learning-based detection in GOOSE communications.

This is an open-access article under the [CC BY-SA](#) license



Keywords:

DoS;
Feature Extraction;
GOOSE;
IEC 61850;
LSTM;

Article History:

Received: February 17, 2026

Revised: July 3, 2026

Accepted: July 14, 2026

Published: October 12, 2026

Corresponding Author:

Tasmi Tasmi
Faculty of Engineering,
Universitas Sriwijaya,
Palembang, Indonesia
Email:
03013622126022@student.unsri.ac.id

INTRODUCTION

The Generic Object-Oriented Substation Event (GOOSE) protocol, which is part of the widely used IEC 61850 standard, enables fast and reliable communication in substations. It

supports the real-time transmission, interoperability, and scalability of Intelligent Electronic Devices (IEDs) [1]. However, GOOSE openness and documented nature make it vulnerable to cyberattacks, including DoS, False

Data Injection (FDI), and Man-in-The-Middle (MiTM) attacks [2][3]. Specifically, Denial of Service (DoS) attacks can cause message delivery delays of over 180 ms, triggering automatic protection failures [4]. This condition endangers the stability of power systems that require millisecond-level responses. Additionally, studies have found that DoS attacks targeting the data-link layer reduce the effectiveness of IEDs, resulting in decreased performance, data loss, and functional failure [5].

Cyberattacks on critical infrastructure are increasing, and Intrusion Detection Systems (IDS) play a key role in protecting networks and systems from threats that could cause physical damage and disrupt operations [6]. These systems can spot and block harmful activity in computer networks [7]. The main goal of IDS is to notice dangerous behavior as it happens [8]. Even though signature-based and rule-based methods are used a lot, both have trouble finding attacks that do not match known patterns [9][10]. Many studies have shown that signature-based IDS fail to detect about 20% of DoS attacks [11][12]. This problem can lead to mistakes where safe traffic is seen as harmful, making it harder for systems to stop threats early [13]. Also, traditional IDS often struggles to keep up with new kinds of attacks [14].

Integrating Machine Learning (ML) algorithms addresses threat detection limitations and enhances IDS accuracy. ML algorithms can discern anomalous network traffic patterns often overlooked by signature-based methods. They can also analyze large volumes of data in real time for immediate processing [15]. Several studies suggest using ML algorithms, such as Random Forest, Decision Tree, and J48, to detect attacks in the IEC 61850 network [16]. However, this approach struggles to handle unique IEC 61850 network features and is suboptimal regarding the extracted attributes. Additionally, studies using Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbor (KNN) for GOOSE message analysis are limited. They rely on non-existent datasets that cannot reflect real-world conditions [17].

Although ML offers solutions, Deep Learning (DL) is more capable of overcoming the weaknesses because it can handle sequential data and complex anomaly patterns that are often difficult to detect by ML methods [18][19]. The Long Short-Term Memory (LSTM) model was used to detect SYN flooding attacks in Transmission Control Protocol (TCP), Manufacturing Message Specification (MMS), and GOOSE traffic [20]. Recurrent Neural Networks (RNNs) exhibited superior performance

in detecting cyber-attacks and achieving high detection accuracy [21]. Other studies have explored using deep learning for anomaly detection in software-defined networking (SDN) with protocol verification and deep learning-based anomaly detection for GOOSE traffic in real time [22]. Despite the proliferation of DL-based IDS studies in IEC 61850, there are several areas that require further refinement, including real-world testing, computational efficiency, IDS model security, and model optimization. Attribute extraction remains a predominant challenge in IDS model studies and optimization because it hinders the detection system's ability to detect highly variable GOOSE traffic [23].

Another problem with intrusion detection in GOOSE is the imbalance in class distribution between normal traffic and attack traffic. This condition biases the model toward the majority class, reducing its detection capability for critical minority classes [24]. In protection systems, failing to detect an attack (false negative) has more serious consequences than other classification errors. Therefore, imbalanced learning is a critical design factor in intrusion detection systems [25].

Many detection systems use machine learning and deep learning. Yet, they often focus only on accuracy metrics. These systems overlook how much model performance can vary with different initialisations and how imbalanced datasets can distort results. Robustness is also lacking because the feature extraction process does not fully use the discriminative power of GOOSE-specific attributes, such as stNum and sqNum.

This study aims to develop an LSTM-based DoS attack detection system using feature extraction. Contributions include: (i) a feature extraction technique using the GOOSE attribute structure to identify DoS attack patterns; (ii) evaluating an LSTM model optimised for balanced data against a baseline without imbalance handling; and (iii) assessing model stability across various random seed values to measure model robustness.

METHOD

The research project was initiated with the creation of a dataset utilizing a laboratory testbed. The testbed under consideration involved the construction of a star topology using an Ethernet switch to facilitate communication between devices. The infrastructure in question comprises an IED publisher, an IED subscriber, a monitoring workstation, and an attacker node. The substation's operational conditions are

realistically represented by an IEC 61850 network, on which all of these are connected. The IED publisher disseminates GOOSE messages, and the IED subscriber receives and processes protection messages instantaneously. The monitoring workstation is utilized for the purposes of configuration and traffic observation during the process of dataset collection. The function of the attacker node is to generate DoS attack traffic on the GOOSE communication network.

In the case of the DoS attack, the perpetrator made four attempts to inundate the network with data. The attacker aimed to disrupt communication between devices and render them unresponsive. The attempts relied on a Python script that transmitted large amounts of data to the network. The following outlines the sequence of events during the attack as executed in the testbed:

- **Attack Initialization:** The configuration process begins by connecting devices to the Ethernet interface, which acts as the gateway to the GOOSE network. This setup creates conditions that allow attacks through message flooding.
- **The following change has been made to the application ID:** The ID used in GOOSE communication must be set to an invalid value (0x03e8). This causes the IED device to receive invalid messages, which may disrupt data flow.
- **Flooding:** Using many attack modules can overwhelm the network. This leads to delays or failures in communication between devices.

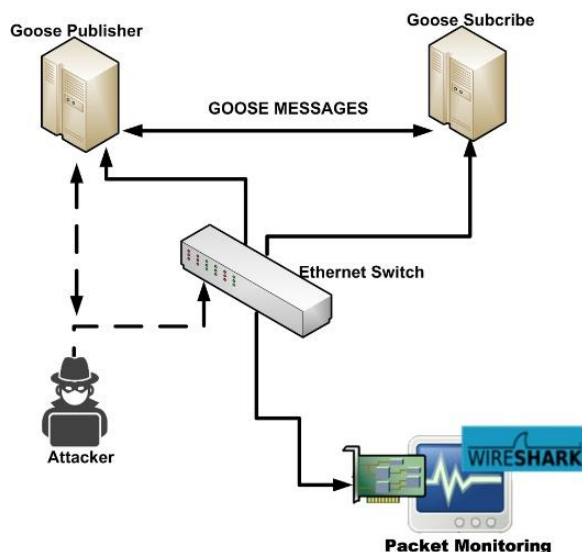


Figure 1. System Under Investigation Topology

Initially, the research team examined the data using Wireshark to validate the dataset and ensure its appropriateness for the study's objectives. This process entailed the verification of the captured packets as being of the GOOSE variety. Secondly, the assignment of labels was based on the experimental scenarios. Thirdly, the study confirmed the presence of features such as stNum, sqNum, ApplID, and other relevant attributes in both the capture and feature extraction results. The dataset produced in the present study has not yet been published in an open repository. However, it is at the discretion of the relevant authorities to grant other researchers access to the aforementioned information, should they wish to verify results, conduct research on the same topic, or further advance research.

In addition to the dataset developed in the laboratory, this study also uses IEC-61850 substation dataset as a benchmark for the proposed model. The present dataset comprises information pertaining to DoS attacks on substations utilising the IEC 61850 protocol [26]. The present study conducted experiments to attack the communication network by flooding traffic in GOOSE. The objective was to exploit the network so that legitimate IEDs would not receive important information, thereby disrupting substation service operations. The details of the dataset are specified in Table 1.

Feature Extraction

This section is a key part of the research. Here, we extract attributes that define network traffic characteristics. This extraction helps in distinguishing legitimate traffic from DoS attack traffic. The extracted attributes include important GOOSE parameters, such as Application Identifier, sqNum, stNum, goose.length, and timeAllowedtoLive. Each parameter helps in identifying normal traffic versus DoS attack traffic.

The data packets are captured by Wireshark, and then the traffic attributes are extracted. Wireshark captures network traffic data in real time, filters and identifies types of packets, and stores the data in PCAP format. PCAP contains a record of network packets detected during the experiment. In the next step, the PCAP file is processed in the attribute extraction stage. After acquiring the PCAP file with GOOSE data, a Python script is used to convert it into Comma-Separated Values (CSV) files. This format is chosen for its enhanced accessibility, interpretability, and analyzability. The CSV data then goes through additional analysis to identify applicable attributes for an attack detection model. These attributes will

undergo further processing to prepare the data for training and testing the detection model.

The selected attributes pertain to GOOSE, encompassing packet identity, communication direction, message size, transmission timing patterns, message sequence, status changes, and data quality. The rationale behind the selection of these particular metrics is twofold: firstly, to ensure optimal accuracy in message delivery, and secondly, to facilitate effective monitoring of status changes. However, there are several attributes that will be removed on the basis that they are constant, redundant, have empty values, or are not directly related to GOOSE communication behavior.

The selection of these attributes is expected to reduce noise, avoid redundancy, and retain features relevant to the GOOSE data. In the subsequent stage of the process, a statistical analysis approach will be employed to select a subset of the set of selected features. This approach is used to determine the strength of each attribute's relationship with the label, and attributes with a strong relationship to the label will be selected for the detection system.

Pre-Processing and Data Balancing

Subsequent to the extraction of salient attributes within the GOOSE package, a series of pre-processing steps is implemented to refine the raw data. The objective of this process is twofold: to enhance the quality and functionality of the data, and to generate data that is prepared for analysis and modeling. Furthermore, the pre-processing phase has been performed to enhance the precision and effectiveness of the model, mitigate bias, and augment the information content of the data. The preprocessing phase is initiated with the inspection of null values, employing the `df.isnull().sum()` function. Subsequently, the data set is examined for redundancy using the `df.duplicated().sum()` syntax, thereby ensuring the model's accuracy.

The subsequent stage of the process is to convert the data types, which were originally categorical, into a numerical format, thereby enabling efficient processing by the deep learning model. This conversion applies to two attributes: `goose.reserve1` and `APPID`. The hexadecimal value of `goose.reserve1` is first normalized by removing the `0x` prefix and then converted to the `int64` data type. For `APPID`, the encoding process uses manual mapping, whereby `0x0000` is mapped to 0, `0x03e8` to 1, and `0x3451` to 2. This method is also used to convert the label feature into a numeric type prior to model training.

Two experiments are used in this study: a baseline and a scenario using data balancing techniques. In the baseline scenario, the model is trained using the original data without class balancing techniques. This is to ensure the data distribution is the same as the initial dataset distribution. This serves as a benchmark to assess how data imbalance affects model performance.

In the second scenario, a data balancing technique was used to address the imbalance between normal traffic and DoS attack traffic. Specifically, Synthetic Minority Over-sampling Technique (SMOTE) was applied to the training dataset to generate synthetic samples for the minority class using linear interpolation based on nearest neighbors. This approach prevented data loss and ensured the validity of the evaluation while allowing examination of the impact of balancing techniques on model performance.

Predictive Modeling and Matrix

The study proposes a new detection system model integrating deep learning methodologies to identify DoS attacks in the GOOSE communication infrastructure. The test system's sequential dataset requires a model capable of capturing temporal dependencies. In GOOSE communication, attributes like `goose.sqNum`, `goose.stNum`, `goose.length`, `frame len`, and `goose.timeAllowedToLive` represents message sequences, status changes, packet sizes, and communication validity time limits. During a DoS attack, these attributes show increased packet volume and traffic changes, potentially disrupting publishers and subscribers.

Consequently, the development of a model that is capable of evaluating individual attributes, in addition to capturing changes in communication, is imperative. Consequently, this study will employ an LSTM model, a specialized type of neural network capable of processing sequential data. Moreover, it is anticipated that this model will exhibit the capacity to retain information from prior communication sequences, thereby facilitating the identification of attack patterns.

The LSTM architecture utilized in this study is depicted in [Figure 2](#), wherein the model receives input from the selection results. The selected attributes are processed through three LSTM layers with 128, 64, and 32 units, respectively, and then fed into a dense layer using the sigmoid function to produce a binary classification. The architecture employed is designed to recognize temporal patterns in GOOSE features, thereby ensuring that the detection process does not rely solely on the

evaluation of individual attribute values. Instead, it is capable of more accurately distinguishing between normal traffic and DoS attacks based on the regularity of sequential communication patterns.

The subsequent phase entailed refining the model configuration through hyperparameter optimization, using a Bayesian optimization approach. The optimal configuration was determined by exploring model parameters, including the number of LSTM layers, the number of neurons in each unit, the activation function, the dropout rate, the optimizer type, the learning rate, and the batch size. The configuration demonstrating optimal performance was identified by achieving the maximum validation F1-score and was subsequently employed as the Optimized LSTM model in the final evaluation process.

The dataset is divided into two parts: 80% for training and 20% for testing. The model uses binary cross-entropy as its loss function to support binary classification, while the Adadelata optimiser controls weight updates during training. The study optimised the LSTM by configuring the number of epochs, batch size, and learning rate, which determine training parameters. Specifically, the experimental setup used 50 epochs and a batch size of 128. To evaluate robustness against variations in random initialization, the model's performance was assessed across ten random seed values: 35, 42, 50, 60, 70, 80, 90, 100, 110, and 120.

The results of this experiment measure the model's performance in terms of accuracy, precision, recall, F1-score, and Area Under the Curve (AUC) metrics to assess the model's effectiveness in distinguishing normal traffic from DoS attacks across various evaluation dimensions. Furthermore, a confusion matrix is utilised to map the results of the classification process and identify patterns of detection errors that arise during the inference process.

Confusion Matrix

The confusion matrix is a statistical tool used to assess the performance of the LSTM-based DoS detection model on the GOOSE communication network. It compares the model's predictions with the actual class labels: DoS traffic (attack) and normal traffic (benign). The confusion matrix consists of four outcomes: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). TP is the correct identification of DoS attacks, FN is DoS traffic incorrectly classified as normal, FP is normal traffic incorrectly labeled as an attack, and TN is the correct identification of benign traffic. These outcomes are crucial for attack detection systems. Accuracy, Precision, Recall, and F1-score can be determined from the confusion matrix using (1)–(4).

1. **Accuracy:** The proportion of accurate predictions made in relation to the total number of data points.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

2. **Precision:** The objective is to ascertain the number of positive predictions that have been accurately determined.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

3. **Recall:** The objective is to ascertain the extent to which positive data is predicted correctly.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

4. **F1-Score:** The utilization of precision and recall in conjunction with each other is essential for providing a more comprehensive evaluation of the performance of classification models.

$$F1 = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (4)$$

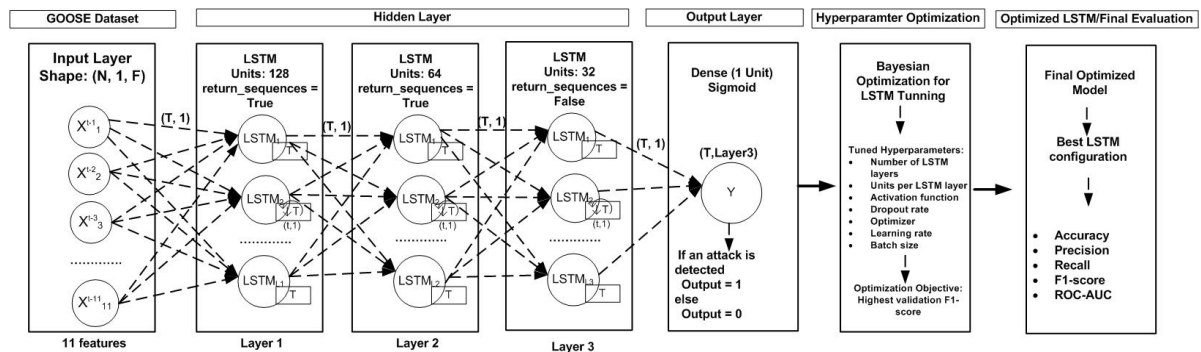


Figure 2. LSTM Architecture

RESULTS AND DISCUSSION

This study involved a series of experiments to comprehensively analyze the behavior of the GOOSE protocol under normal and attack traffic conditions. The study is divided into four sections. The steps taken were: (1) data collection, (2) feature extraction, (3) dataset preprocessing and balancing, and (4) predictive modeling and performance evaluation.

Data Collection

The initial stage of this research was the construction of an IEC 61850 testbed. The system comprised a GOOSE Publisher, a GOOSE Subscriber, a switch, a packet capture, and an attacker. The DoS attack utilized a script to inundate the system. It disseminated fabricated GOOSE messages with ApplID 0x03e8. The objective was to disrupt real-time communication between IEDs. The aim was not to modify or replace the GOOSE structure itself. Instead, the attack leveraged data-link layer vulnerabilities to gain network access.

The results of the packet capture demonstrate that the attack had two principal outcomes: message delivery times became erratic, and the data volume and magnitude increased dramatically. This phenomenon is substantiated by the observation that the inter-arrival time (frame.time_delta) has undergone a substantial decrease, with an average reduction from 1000 milliseconds to less than 1 millisecond. The rate of data transmission exhibited an increase of 24.64 frames per second. The goose.length value exhibited erratic fluctuations, falling within the 100–200-byte range under stable or normal conditions. However, in this experiment, the attacker transmitted substantial data packets measuring 1.5 kilobytes (KB) and 2.6 KB.

The findings show a twofold increase in GOOSE data, indicating data injection flooding, which can cause message delivery delays of over 180 milliseconds. Such delays can trigger relay protection failures in Supervisory Control and Data Acquisition (SCADA) systems. GOOSE attributes stNum and sqNum continue to yield stNum ≥ 1 values and an increase. This fact shows the attack does not involve protocol manipulation, but rather increases the network packet rate, inter-arrival time, and payload size. The results on the communication protocol used in the experiment showed that GOOSE traffic was 98% of the observed traffic. The result shows the dataset accurately captures traffic on the IEC 61850 network, excluding TCP or User Datagram Protocol (UDP) traffic on a public network.

The processed dataset will be used to train the model. This process yielded 39,840 records, including 4,565 normal and 35,275 DoS attacks, with attacks accounting for 88.54% of the total traffic. The public dataset yields 49,493 records, with a lower number of DoS data 10,000 and 39,493 normal records, resulting in an attack proportion of 20.20%. Table 1 compares the data distributions across both datasets, which serve as a simulation illustrating communication within GOOSE, providing a foundation for training, evaluation, and the performance of the LSTM model.

A study of the public dataset was conducted [27], and the results of the analysis are as follows: of the total of 49,493 data; 84.45% were GOOSE, 5.87% were mDNS, 5.69% were ARP, 2.733% were ICMPv6, 1.19% were DHCP, and 0.044% were browser-related. This composition indicates that GOOSE traffic dominates the public dataset.

Feature Extraction

The objective of the attribute extraction stage in this study is to identify the main GOOSE protocol attributes that directly reflect the impact of DoS attacks on GOOSE communication. Due to the network's inherent hierarchical structure and packet encapsulation, deep learning models cannot use raw data in PCAP format directly for training. Coding was developed to convert PCAP files into a structured CSV format. This approach generates GOOSE-specific attributes, such as goose.stNum, goose.sqNum, frame.time_delta, and goose.length. These attributes will be used to detect DoS attack patterns. The results from this process will also be used for pre-processing and LSTM model training.

This stage yielded 148,438 rows of data, comprising 236 initial attributes, which encompassed various layers of the network protocol. In consideration of the exclusive focus of this study on GOOSE, an attribute selection process was conducted for the purpose of identifying attributes associated with GOOSE communication characteristics, encompassing both normal and DoS data

Table 1. Distribution of DoS attack datasets in GOOSE communication

Dataset	Total Records	Attack Records	Normal Records	Attack Proportion (%)
Internal Dataset	39,840	35,275	4,565	88.54
IEC-61850 Substation Dataset	49,493	10,000	39,493	20.20

. The results of the preliminary selection and evaluation process identified 17 attributes that characterize GOOSE communication, as illustrated in Table 2. Subsequent analysis of these 17 attributes revealed their composition of three distinct data types: floating-point numeric, integer numeric, timestamp, categorical, and boolean/binary. Subsequently, attributes with categorical, timestamp, and boolean data types will undergo conversion to a numerical format, thus enabling processing by the detection system.

Table 2. The Presentation of Attribute Feature Extraction Results

Attribute	Description
frame.time_delta	Time interval between GOOSE packets (in seconds); near-zero values indicate flooding behavior.
frame.time_relative	The temporal trend analysis employs the packet arrival time relative to the initiation of capture.
goose.t	GOOSE event timestamp; sudden changes signal unusual activity.
goose.timeAllowedToLive	The maximum allowed time before the next packet is expected. Violations suggest communication disruption.
goose.stNum	Status number increments when data set values change and remains constant (typically = 1) during valid DoS flooding.
goose.sqNum	The sequence number increases with each retransmission, and an exponential increase indicates flooding..
confRev	Revision of dataset configuration may indicate manipulation.
AppID	16-bit Application Identifier
goID	GOOSE message sender identifier (IED name) validates message origin.
frame.len	The total Ethernet frame length (in bytes) may indicate an abnormal network load if it increases suddenly.
goose.length	GOOSE payload length (in bytes) is utilized to identify anomalous message sizes.
goose.numDataSetEntries	The number of entries in the GOOSE dataset reflects content manipulation.
frame.number	This sequence number detects packet duplication or loss..
allData_bool_count	Number of boolean values in allData; DoS patterns often remain the same.
allData_int_mean	The mean of the integer values in allData indicates message stability.
goose.simulation	This is a flag indicating simulation mode. Use it only in simulation contexts.
goose.reserve1.s_bit	Status bit in reserved field. Unexpected toggling may indicate illicit modification.

These attributes were then divided into three groups based on the differences in behavior between normal GOOSE communication and DoS attacks.

1. Temporal attributes (such as timing and frequency) and DoS patterns in GOOSE messages, identified by the AppID 0x03e8, are transmitted at an average interval of 0.850 ms. The minimum interval is 0.050 ms, equating to 20,000 packets per second. Standard GOOSE traffic patterns demonstrate an average interval of 1002.4 ms, with a range from 998.1 to 1005.7 ms. These intervals match the typical characteristics of IEC 61850-based (an international standard for substation automation) substations.
2. State Protocol Attributes identify messages marked as attacks when they use gocbRef=LIED10CTRL/LLN0\$Status and Application 0x03e8. The investigation shows an attack pattern that mimics valid IED devices. Key attributes such as goose.stNum stays at 1, and goose.sqNum remains in the message. There is no evidence of GOOSE structure exploitation.
3. The payload attribute is defined as a message identified as paThe payload attribute is defined as a message identified as part of an attack that produces a variability value between 48 and 664 bytes in the goose.length attribute and a value in goose.num, DataSetEntries of 1 to 5 entries. These values are indicative of changes in payload size as volumetric anomalies in DoS detection.

Figure 3 shows the validation results, which reveal a clear correlation between the raw data and the extracted results. The consistency of these results indicates that feature extraction preserves important information from GOOSE packets.

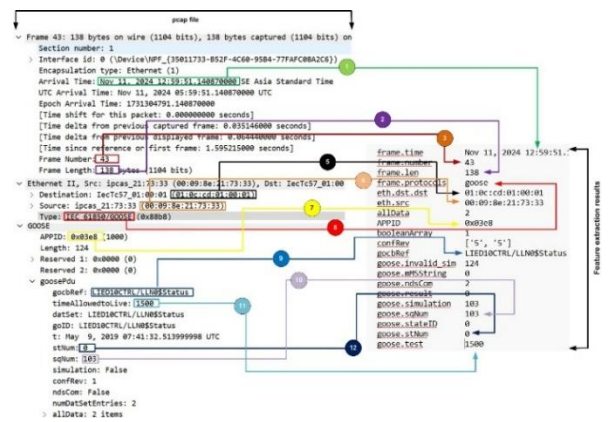


Figure 3. Validate Raw Data and Extract Features

These extraction results do not lose important features such as `frame.time` and `frame.len`, which represent time and packet size information, while `AppID`, `stNum`, and `sqNum` relate to application identity, status changes, and message sequencing in GOOSE communication. The validation results are crucial because the attributes used in training and testing the LSTM model have been thoroughly verified against GOOSE. Thus, the results of this stage yield attributes that align with the characteristics and operational requirements for attack detection.

Pre-processing and Data Balancing

To produce valid data for attack detection using DL, several stages are carried out in the data preparation process. First, determine the extent of data loss resulting from DoS attacks. Second, identify message duplication caused by the retransmission mechanism in GOOSE communication. Third, normalize data to address attribute scale differences. Fourth, address the imbalance between normal and DoS classes.

The initial stage was a thorough examination of the dataset, which revealed 87 attributes with empty values, including `goose.reserve1` and `goose.simulation`, while key attributes such as `goose.stNum`, `goose.sqNum`, `frame.time_delta`, and `goose.length` were not missing. In the second stage, the attack duplicated data to show the attack was only sending high-volume messages; however, no data duplication was found, possibly because the attack did not use a replay attack, and the feature extraction process did not result in duplication. In the third stage, min-max scaling was applied to features, scaling them between 0 and 1 while preserving their original distribution to better detect real-time protocol anomalies.

The fourth stage involves identifying class imbalance, as evidenced by the testbed experiment results for the dataset, which revealed 88.54% of the sample as DoS attacks and 11.46% as normal activity. If not addressed, this imbalance leads the model to prioritize detecting the majority class, which can result in false negatives in substation protection. The implementation of SMOTE balancing was motivated by its capacity to specifically improve IDS model performance on key metrics, such as accuracy, precision, recall, and F1 score, during subsequent evaluations [28][29].

The SMOTE outcome is a balanced dataset with 50% of samples in each of the following categories: The total number of DoS and normal data records is 35,275. It is important to note that these results do not stem from technical processes; rather, they are authentic

representations of realistic DoS attacks. The dataset is prepared for training deep learning models for detection. This approach facilitates a more comprehensive evaluation of recall outcomes and false negative rates.

Predictive Modeling and Performance Evaluation

Predictive Modeling Without Optimization

A series of ten experiments was conducted to assess the performance and stability of the LSTM model. A series of variations of the `random_state` parameter were contemplated, as well as 11 GOOSE features, including `frame.time_delta`, `goose.length`, `goose.stNum`, and `goose.sqNum`. The data ratio was set at 80:20, and the objective was to assess the model's capacity to withstand random variations arising from the division of data between the training and testing sets [30].

As shown in Table 3, the LSTM model was first tested against a baseline without data balancing, then compared with GRU, CNN, and Transformer models. The LSTM model consistently produced stable results in terms of accuracy, recall, average F1 score, precision, and AUC. The model's average accuracy was $98.59\% \pm 0.78\%$, average recall was $98.37\% \pm 0.75\%$, the average F1 score was $98.59\% \pm 0.77\%$, the average precision was $98.82\% \pm 0.82\%$, and the AUC was $99.49\% \pm 0.30\%$. These results show that `random_state` set to 80 yielded optimal parameters, achieving an accuracy of 99.22%, a recall of 98.92%, and an AUC of 98.99%. These findings reinforce the model's effectiveness in distinguishing between normal traffic and DoS attacks.

It is evident from the test data that the LSTM model produced elevated, consistent evaluation scores. This is shown by the high accuracy, F1-score, and AUC metrics, as well as its performance across various `random_state` configurations. This stability suggests that the model performs well under a specific data split scenario and maintains performance across different data split variations. Conversely, GOOSE communication data exhibits temporal patterns that the LSTM can use to learn sequential relationships between messages.

The employment of 11 GOOSE attributes in conjunction with the imbalance technique facilitated the model's attainment of a more balanced representation between the normal and attack classes. Consequently, it can be deduced that the model's high performance can be attributed to three primary factors: the quality of the features employed, a more representative data distribution, and the LSTM's capacity to

extract temporal dependencies in GOOSE communications.

This experiment compared the performance of the LSTM, CNN, GRU, and Transformer models under the same conditions. Table 4 shows the results. The GRU model performed slightly better than the LSTM model. This is because GRU is more compact than LSTM [31]. However, LSTM performed competitively compared to CNN and Transformer, reinforcing its strong overall performance.

Predictive Modeling with Hyperparameter Optimization

Despite the negligible disparities in performance between LSTM and GRU, the findings suggest that LSTM has not yet attained optimal performance. Consequently, the research proceeded with hyperparameter optimization to explore enhanced LSTM configurations. This process yielded a more streamlined architecture, comprising a single-layer LSTM with 160 units, a ReLU activation function, a dropout rate of 0.131980, the RMSprop optimizer, a learning rate of 0.0001, and a batch size of 64.

To assess the optimization results, we re-examined the LSTM model using the same experimental paradigm. This re-examination compared the LSTM model with the GRU, CNN, and Transformer models. The findings indicated that the LSTM model performed better than the three alternative models. As shown in Table 5, the LSTM model demonstrated a significant accuracy of 0.9976 ± 0.0008 , precision of 0.9984 ± 0.0016 , recall of 0.9969 ± 0.0022 , F1-score of 0.9976 ± 0.0008 , and AUC of 1.0000 ± 0.0000 . These results indicate a notable improvement compared to the preliminary experimental findings, particularly regarding the F1-score and AUC metrics.

The findings indicate that hyperparameter optimization substantially enhances the model's capacity to discern attack patterns in GOOSE. They also suggest that LSTM performance is not determined solely by architectural complexity, but is strongly influenced by hyperparameter configurations that match the data characteristics.

Table 3. Validation Results with Random State

Random State	Accuracy	Recall	F1	Precision	AUC-ROC
35	0.9816	0.9671	0.9813	0.9959	0.9975
42	0.9884	0.9809	0.9883	0.9958	0.9937
50	0.9918	0.9900	0.9919	0.9938	0.9978
60	0.9905	0.9904	0.9906	0.9908	0.9954
70	0.9664	0.9821	0.9668	0.9519	0.9975
80	0.9922	0.9892	0.9922	0.9952	0.9899
90	0.9823	0.9762	0.9821	0.9881	0.9900
100	0.9892	0.9863	0.9892	0.9921	0.9972
110	0.9859	0.9849	0.9860	0.9871	0.9941
120	0.9905	0.9902	0.9905	0.9908	0.9961
mean $\pm$ std	0.9859 ± 0.0077	0.9837 ± 0.0075	0.9859 ± 0.0077	0.9881 ± 0.0131	0.9949 ± 0.0029

Table 4. Results of the Baseline and Proposed Models on the Internal Dataset and Public Dataset

Dataset	Model	Accuracy	Precision	Recall	F1-Score	AUC
Internal Dataset	Baseline	0.9678 ± 0.0173	0.9760 ± 0.0148	0.9881 ± 0.0150	0.9819 ± 0.0098	0.9390 ± 0.0422
	Proposed	0.9859 ± 0.0077	0.9881 ± 0.0131	0.9837 ± 0.0075	0.9859 ± 0.0077	0.9949 ± 0.0029
IEC-61850 Substation Dataset	Proposed	0.9429 ± 0.1563	0.9497 ± 0.1587	0.9864 ± 0.0049	0.9939 ± 0.0032	0.9959 ± 0.0014
GRU		0.9898 ± 0.0004	0.9918 ± 0.0021	0.9915 ± 0.0000	0.9916 ± 0.0011	0.9982 ± 0.0016
CNN		0.9668 ± 0.0131	1.0000 ± 0.0000	0.9334 ± 0.0262	0.9654 ± 0.0140	0.9940 ± 0.0003
Transformer		0.8828 ± 0.0004	0.9920 ± 0.0023	0.7710 ± 0.0028	0.8676 ± 0.0009	0.8704 ± 0.0038

Table 5. Hyperparameter Optimization Results and Performance Comparison of Deep Learning Models

Model	Accuracy	Precision	Recall	F1-Score	ROC_AUC
Optimized LSTM - Proposed	0.9976 ± 0.0008	0.9984 ± 0.0016	0.9969 ± 0.0022	0.9976 ± 0.0008	1.0000 ± 0.0000
GRU	0.9929 ± 0.0063	0.9927 ± 0.0104	0.9933 ± 0.0027	0.9930 ± 0.0061	0.9962 ± 0.0076
CNN	0.9328 ± 0.0639	0.9996 ± 0.0012	0.8657 ± 0.1288	0.9227 ± 0.0818	0.9862 ± 0.0216
Transformer	0.8841 ± 0.0023	0.9930 ± 0.0008	0.7739 ± 0.0051	0.8698 ± 0.0031	0.8719 ± 0.0029

Results of the Statistical Validity Test

Hyperparameter optimization significantly improved the LSTM model's performance. However, statistical testing is necessary to validate this improvement. The objective is to confirm that the observed performance difference is not due to data splitting variations or the use of the `random_state` parameter. Instead, it should appear as a consistent and reproducible difference across repeated tests.

Table 6 shows differences in Accuracy, F1-score, and ROC-AUC among the evaluated models. These results indicate that performance variations among models are influenced by random factors and model architecture in learning GOOSE patterns. Hyperparameter optimization can yield consistency and statistical significance.

The Wilcoxon signed-rank test shows significant performance differences among the models. It also indicates that the Optimized LSTM consistently outperforms the CNN, GRU, and Transformer across all evaluation metrics. These results show that the performance improvement following hyperparameter optimization is consistent across all model comparisons.

As demonstrated in Table 7, the Optimized LSTM model exhibited superior performance

compared with the other models under consideration. For accuracy, it performed significantly better than the CNN, GRU, and Transformer models, with adjusted p-values of 0.0059, 0.0176, and 0.0059, respectively. Similar findings were observed for the F1-score metric, where the Optimized LSTM showed marked superiority, with adjusted p-values of 0.0059 against the CNN, 0.0117 against the GRU, and 0.0059 against the Transformer. Comparable outcomes were also obtained for the ROC-AUC metric, with Optimized LSTM achieving notably higher performance than all three comparison models, each with an adjusted p-value of 0.0059.

The adjusted p-values obtained were all below the significance threshold of 0.05, indicating that the improved performance of the Optimized LSTM is supported by strong statistical evidence, rather than by higher evaluation metrics alone. These findings suggest that the observed enhancement in performance is not attributable to random variations during the testing process. Rather, it is a consistent outcome of hyperparameter optimization. Consequently, hyperparameter optimization enhances the LSTM model's capacity to differentiate between normal traffic and DoS attacks on GOOSE.

Table 6. Results of the Friedman Test

Metric	Compared Models	Statistic	p-value	p-value adjusted	Interpretation
Accuracy	CNN, Optimized LSTM, GRU, Transformer	265.200	<0.0001	-	Significant difference among models
F1-score	CNN, Optimized LSTM, GRU, Transformer	265.200	<0.0001	-	Significant difference among models
ROC-AUC	CNN, Optimized LSTM, GRU, Transformer	289.200	<0.0001	-	Significant difference among models

Table 7. Results of the Statistical Test Using the Wilcoxon Signed-Rank Test

Metric	Compared Models	Statistic	p-value	p-value adjusted	Interpretation
Accuracy	Optimized LSTM vs CNN	0.0000	0.0020	0.0059	Optimized LSTM significantly higher than CNN
Accuracy	Optimized LSTM vs GRU	15.0000	0.0059	0.0176	Optimized LSTM significantly higher than GRU
Accuracy	Optimized LSTM vs Transformer	0.0000	0.0020	0.0059	Optimized LSTM significantly higher than Transformer
F1-score	Optimized LSTM vs CNN	0.0000	0.0020	0.0059	Optimized LSTM significantly higher than CNN
F1-score	Optimized LSTM vs GRU	10.0000	0.0039	0.0117	Optimized LSTM significantly higher than GRU
F1-score	Optimized LSTM vs Transformer	0.0000	0.0020	0.0059	Optimized LSTM significantly higher than Transformer
ROC-AUC	Optimized LSTM vs CNN	0.0000	0.0020	0.0059	Optimized LSTM significantly higher than CNN
ROC-AUC	Optimized LSTM vs GRU	0.0000	0.0020	0.0059	Optimized LSTM significantly higher than GRU
ROC-AUC	Optimized LSTM vs Transformer	0.0000	0.0020	0.0059	Optimized LSTM significantly higher than Transformer

Results of the Training Time Analysis and Inference

This study measures computational efficiency to evaluate model performance and assess the feasibility of applying the models to an attack detection system for GOOSE communications. Table 8 compares training time, inference time, and inference time per sample for CNN, Optimized LSTM, GRU, and Transformer based on ten test scenarios. These results show the trade-off between detection capability and computational requirements for each model.

The evaluation of computation time revealed that the CNN model exhibited optimal training efficiency, with an average training duration of 91.4296 ± 2.4645 seconds. Conversely, the Optimized LSTM model necessitated a more protracted training period, with an average duration of 265.2818 ± 13.7208 seconds. This difference reflects the recurrent nature of LSTMs and the greater complexity of the optimization process compared to convolution-based models. However, the Optimized LSTM model exhibited efficient inference performance, with an inference time of 1.6103 ± 0.4887 seconds and an average processing time per sample of 0.1141 ± 0.0346 milliseconds. These results are slightly lower than those of the GRU, which achieved 0.1678 ± 0.0304 ms/sample, and the Transformer, which achieved 0.1435 ± 0.0330 ms/sample.

The results of the computational time analysis show that the Optimized LSTM strikes a good balance between detection performance and computational efficiency. It requires a longer training time than some benchmark models, but it offers slightly better detection capabilities while maintaining an acceptable inference time for the GOOSE traffic monitoring application.

Analysis Curve AUC and Confusion Matrix

Figure 4 shows the results of the model accuracy test for random_state = 80. These results illustrate the generalization ability of the LSTM model. The accuracy curve increases significantly between epochs 0 and 10, demonstrating the model's ability to recognize data patterns.

Table 8. Training and Inference Times

Model	Training Time (s)	Inference Time (s)	Inference Time per Sample (ms)
CNN	91.4296 ± 2.4645	1.1010 ± 0.2755	0.0780 ± 0.0195
Optimized LSTM	265.2818 ± 13.7208	1.6103 ± 0.4887	0.1141 ± 0.0346
GRU	227.7177 ± 9.3433	2.3672 ± 0.4293	0.1678 ± 0.0304
Transformer	205.8239 ± 8.0118	2.0254 ± 0.4661	0.1435 ± 0.0330

Between epochs 10 and 50, the training and validation accuracy curves stabilize around 1.00 with no significant difference. These results also indicate that overfitting did not occur, as evidenced by the closeness of the two curves. Furthermore, the model did not suffer from underfitting, as it was able to learn data patterns and achieve high performance on both the training and validation data.

Figure 5 shows the loss curve used to track changes in loss during training. The results from 50 epochs show a rapid decrease from epochs 0 to 10, indicating that the model parameters are adjusting more effectively to reduce error. From epoch 10 to 50, both curves gradually decline, with only a slight difference as they approach the minimum value. The test results show that the LSTM model architecture effectively minimizes detection error. These results indicate no significant increase in validation loss compared to training loss, meaning the model does not exhibit overfitting. Meanwhile, the loss curve continues to decline steadily until it reaches a low value; this demonstrates that the model is capable of learning GOOSE patterns and does not suffer from underfitting.

Figure 6 shows that all ROC curves are close to the upper left corner, indicating a high True Positive Rate and low False Positive Rate. The AUC values produced in each random state are ≥ 0.98 , varying only slightly between 0.9899 and 0.9978, confirming model stability, generalization, and the ability to distinguish between normal and DoS classes. These results also demonstrate the robustness of the LSTM architecture against variations in random state.

The confusion matrix results are presented in Figure 7 for 14,110 samples. Of these, 7,018 were identified as normal samples, and 6,983 were identified as attack samples. Thirty-three samples were identified as false positives, meaning that normal samples were classified as attacks. Conversely, 76 samples were classified as false negatives, meaning attack samples were not classified as normal. This resulted in a false negative rate (FNR) of 1.08% and a false positive rate (FPR) of 0.47% (33/7.051). This model is effective in detecting attacks while exhibiting a low false alarm rate. However, it is important to note that undetected attacks can cause service disruptions and system availability issues.

The present study aims to evaluate the model that has been utilised. To this end, a comparison will be made between the proposed model and a baseline model. The baseline model is an LSTM model that has been trained without the application of data imbalance handling techniques. Table 4.

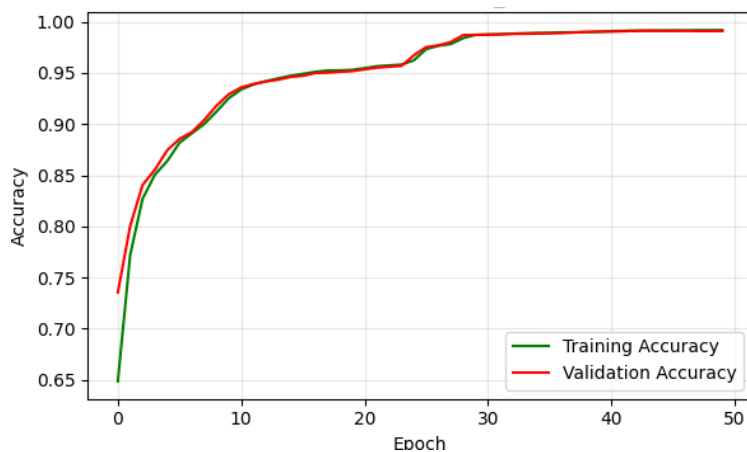


Figure 4. Learning Curve of Accuracy

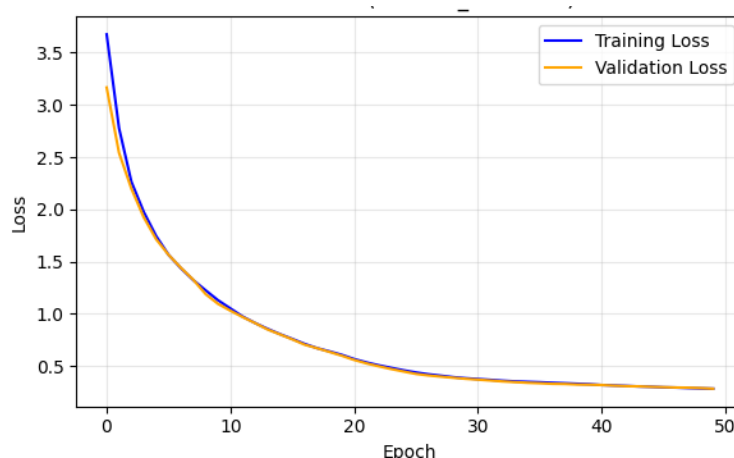


Figure 5. Learning Curve of Loss

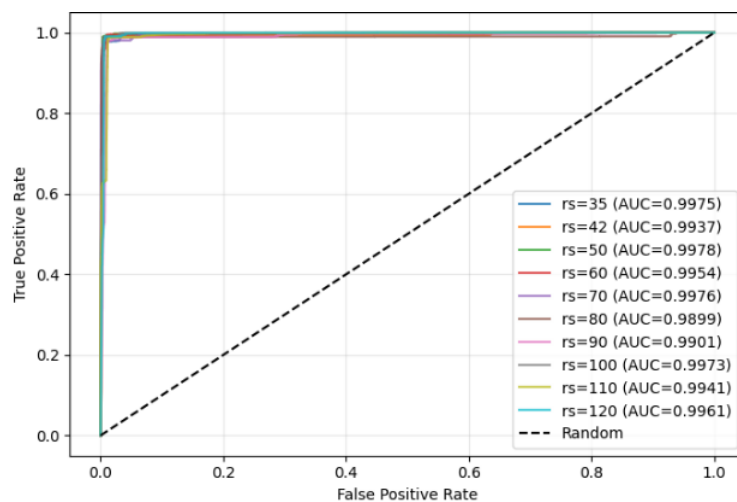


Figure 6. AUC Curve

The model using the balancing technique outperformed the baseline on the internal dataset. The proposed model achieves an accuracy of 0.9859 ± 0.0077 , precision of 0.9881 ± 0.0131 , recall of 0.9837 ± 0.0075 , an F1-score of 0.9859 ± 0.0077 , and an AUC of 0.9949 ± 0.0029 . In

contrast, the baseline yields an accuracy of 0.9678 ± 0.0173 , precision of 0.9760 ± 0.0148 , recall of 0.9881 ± 0.0150 , F1-score of 0.9819 ± 0.0098 , and AUC of 0.9390 ± 0.0422 . These results indicate improved precision, F1-score, and stability in the proposed model.

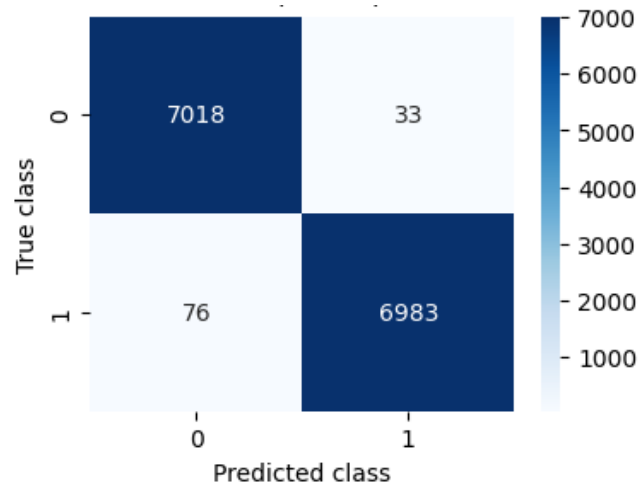


Figure 7. Confusion Matrix on Random State 80

Table 9. Comparison of Previous Deep Learning-Based Approaches for Attack Detection in IEC 61850 GOOSE Communication

Study	Attack Type	Method	Main Result
A. Senel [32]	DoS, replay, and masquerade attacks	CNN, LSTM, and hybrid CNN-LSTM	Hybrid CNN-LSTM 99.72% accuracy for multi-class attack detection.
T. Fernando [33]	GOOSE spoofing	DNN, LSTM, and autoencoder-assisted anomaly detection	LSTM trained 99.47% training accuracy and 99.68% validation/test accuracy. only GOOSE features, the 97.85%.
J. A. de Oliveira [34]	Masquerade attack	Decision Tree, Random Forest, RNN, LSTM, BiLSTM, GRU, and BiGRU	GRU and RNN F1-score of 97.96%, accuracy 99.94%, recall of 100%, and precision of 96%. multi-class GRU F1-score and accuracy of 99.97%.
Present study	DoS attack	GOOSE-specific feature extraction, feature selection, based balancing, Optimization Hyperparameter LSTM, CNN, GRU, and Transformer comparison	The proposed Optimization LSTM a mean accuracy of $99.76\% \pm 0.08\%$, precision of $99.84\% \pm 0.16\%$, recall of $99.69\% \pm 0.22\%$, F1-score of $99.76\% \pm 0.08\%$, and AUC of $100\% \pm 0.000\%$ on the internal dataset.

The model was also tested on a public dataset to validate the results. The accuracy, precision, recall, and AUC were 0.9429 ± 0.1563 , 0.9497 ± 0.1587 , 0.9864 ± 0.0049 , and 0.9959 ± 0.0014 , respectively.

The internal dataset had high recall and AUC, and the public dataset had lower accuracy and precision. The model is effective and generalises well on public data. Although real-time testing has not been conducted, results from internal and public datasets indicate potential for deployment outside the lab setting. This is supported by high performance on internal datasets, generalisation to public datasets, and stable results across random seed variations

Comparison with Previous Studies

The model utilized in this study was evaluated through a comparative analysis with several prior studies that employed deep learning techniques. As demonstrated in Table 9, previous studies have effectively demonstrated that deep learning models can achieve high detection performance with an accuracy of over 97%. However, the key takeaway is that the extant

literature uses disparate datasets, attack types, feature representations, and experimental configurations, which limits direct comparison.

The Optimized LSTM model in this study achieved 99.76% accuracy, an F1-score of 99.76%, and a precision of 99.84% for DoS detection in GOOSE communications. Previous studies have produced high accuracy values, but this study stands out through GOOSE-specific features, feature selection, class imbalance handling, LSTM hyperparameter optimization, and performance validation. The approach is expected to maintain high performance by focusing on the temporal characteristics of GOOSE traffic to detect DoS attacks.

Limitations and Future Development

The proposed Optimized LSTM model has demonstrated the capacity to produce satisfactory outcomes in the detection of DoS attacks in GOOSE communications, but this study has limitations. Firstly, it was conducted on only one type of attack DoS precluding a comprehensive evaluation of other attack types.

Secondly, the evaluation of this model used a GOOSE traffic dataset obtained from a simulated environment, thereby precluding its ability to adequately represent the conditions at a substation with regard to IED configurations, changes in network conditions, communication disruptions, and operational loads. Additionally, while the proposed model demonstrates low inference time, it has not yet been tested directly in a substation environment.

In the context of future research endeavors, the proposed model can serve as a framework for evaluating cyberattacks on GOOSE communications. Its multi-class scenarios can improve generalizability and expand its capacity to counter a broader range of threats. Furthermore, the model can be developed using larger-scale datasets, including traffic from real-world substation environments. Subsequent research may entail integrating an adaptive learning methodology to facilitate real-time intrusion detection systems.

CONCLUSIONS

This research developed a DoS attack detection system for GOOSE communications using the Optimized LSTM approach. The model evaluation results indicate optimal attack detection capabilities, with average accuracy of $99.76\% \pm 0.08\%$, precision of $99.84\% \pm 0.16\%$, recall of $99.69\% \pm 0.22\%$, F1-score of $99.76\% \pm 0.08\%$, and ROC-AUC of 1.0000 ± 0.0000 . These results were achieved by combining GOOSE-specific feature extraction, feature selection, balancing techniques, and LSTM hyperparameter optimization. Statistical analysis also found that Optimized LSTM produced a significant performance difference compared with CNN, GRU, and Transformer models. Tests across several random states showed that the model maintained its detection capability. Although the Optimized LSTM requires a higher training time, it demonstrates efficient inference time, making it applicable as an IDS mechanism in GOOSE communications. However, the study has limitations: testing was conducted on DoS attacks with a testbed dataset. Further research should evaluate other cyberattacks using a wider-scope dataset and validate the model on a real substation system to improve generalizability and support the implementation of a real-time detection system.

ACKNOWLEDGMENT

The author would like to express their gratitude to the promoter and co-promoter for their guidance and support. Additionally, gratitude is extended to Sriwijaya University and Indo Global Mandiri University for their provision of facilities and resources that facilitated this research. The author would also like to acknowledge the contributions of

research colleagues at comnet who assisted in data collection. Additionally, gratitude is extended to family members and all individuals.

REFERENCES

- [1] H. T. Reda *et al.*, "Vulnerability and impact analysis of the iec 61850 goose protocol in the smart grid," *Sensors*, vol. 21, no. 4, pp. 1–20, 2021, doi: 10.3390/s21041554.
- [2] M. Boeding, M. Hempel, H. Sharif, J. Lopez, and K. Perumalla, "A flexible OT testbed for evaluating on-device implementations of IEC-61850 GOOSE," *Int. J. Crit. Infrastruct. Prot.*, vol. 42, Art. no. 100618, 2023, doi: 10.1016/j.ijcip.2023.100618.
- [3] S. M. S. Hussain *et al.*, "A novel hybrid methodology to secure GOOSE messages against cyberattacks in smart grids," *Sci. Rep.*, vol. 13, no. 1, pp. 1–13, 2023, doi: 10.1038/s41598-022-27157-z.
- [4] S. Ashraf, M. H. Shawon, H. M. Khalid, and S. M. Muyeen, "Denial-of-service attack on iec 61850-based substation automation system: A crucial cyber threat towards smart substation pathways," *Sensors*, vol. 21, no. 19, 2021, doi: 10.3390/s21196415.
- [5] X. Cai, K. Shi, K. She, S. Zhong, Y. Soh, and Y. Yu, "Performance Degradation Estimation Mechanisms for Networked Control Systems Under DoS Attacks and its Application to Autonomous Ground Vehicle," *IEEE Trans. Cybern.*, vol. 54, no. 5, pp. 2992–3002, 2024, doi: 10.1109/TCYB.2023.3286878.
- [6] T. Sowmya and E. A. Mary Anita, "A comprehensive review of AI based intrusion detection system," *Meas. Sensors*, vol. 28, no. June, p. 100827, 2023, doi: 10.1016/j.measen.2023.100827.
- [7] L. Nkenyereye, B. A. Tama, and S. Lim, "A stacking-based deep neural network approach for effective network anomaly detection," *Comput. Mater. Contin.*, vol. 66, no. 2, pp. 2217–2227, 2021, doi: 10.32604/cmc.2020.012432.
- [8] P. V. Pramila and M. Gayathri, "Analysis of Accuracy in Anomaly Detection of Intrusion Detection System Using Naïve Bayes Algorithm Compared Over Gaussian Model," *ECS Trans.*, vol. 107, no. 1, pp. 13977–13991, 2022, doi: 10.1149/10701.13977ecst.
- [9] A. H. Salem, S. M. Azzam, O. E. Emam, and A. A. Abohany, "Advancing cybersecurity: a comprehensive review of AI-driven detection techniques," *Journal of Big Data*, vol. 11, no. 1, p. 105, 2024, doi: 10.1186/s40537-024-00957-y.
- [10] A. H. Ali *et al.*, "Unveiling machine learning strategies and considerations in intrusion

- detection systems: a comprehensive survey," *Front. Comput. Sci.*, vol. 6, 2024, doi: 10.3389/fcomp.2024.1387354.
- [11] M. Dimolianis, A. Pavlidis, and V. Maglaris, "Signature-based traffic classification and mitigation for ddos attacks using programmable network data planes," *IEEE Access*, vol. 9, pp. 113061–113076, 2021, doi: 10.1109/ACCESS.2021.3104115.
- [12] M. Raman, C. M. Ahmed, and A. Mathur, "Machine learning for intrusion detection in industrial control systems: challenges and lessons from experimental evaluation," *Cybersecurity*, vol. 4, no. 1, p. 27, 2021, doi: 10.1186/s42400-021-00095-5.
- [13] H. Attou *et al.*, "Towards an Intelligent Intrusion Detection System to Detect Malicious Activities in Cloud Computing," *Appl. Sci.*, vol. 13, no. 17, 2023, doi: 10.3390/app13179588.
- [14] W. Li, W. Meng, and L. F. Kwok, "Surveying Trust-Based Collaborative Intrusion Detection: State-of-the-Art, Challenges and Future Directions," *IEEE Commun. Surv. Tutorials*, vol. 24, no. 1, pp. 280–305, 2022, doi: 10.1109/COMST.2021.3139052.
- [15] S. Singhal and P. Yadav, "Machine Learning Techniques for Intrusion Detection System: A Survey," *Lect. Notes Electr. Eng.*, vol. 907, pp. 29–40, 2022, doi: 10.1007/978-981-19-4687-5_3.
- [16] V. E. Quincozes, S. E. Quincozes, C. Albuquerque, D. Passos, and D. Mosse, "Feature Extraction for Intrusion Detection in IEC-61850 Communication Networks," in *2022 6th Cyber Security in Networking Conference (CSNet)*, pp. 1–7, 2022, doi: 10.1109/CSNet56116.2022.9955599.
- [17] T. S. Ustun *et al.*, "Machine learning-based intrusion detection for achieving cybersecurity in smart grids using IEC 61850 GOOSE messages," *Symmetry (Basel)*, vol. 13, no. 5, p. 826, 2021, doi: 10.3390/sym13050826.
- [18] P. Chiranjeevi, Yadavalli Ramya, Chinthala Balaji, Bathini Shashank, and Abbdai Sainath Reddy, "Uncovering time series anomaly using deep learning technique," *World J. Adv. Res. Rev.*, vol. 22, no. 1, pp. 879–887, 2024, doi: 10.30574/wjarr.2024.22.1.1129.
- [19] S. Bakker, Y. Ma, and S. S. M. Ziabari, "Exploring a Hybrid Deep Learning Approach for Anomaly Detection in Mental Healthcare Provider Billing: Addressing Label Scarcity through Semi-Supervised Anomaly Detection," *arXiv preprint arXiv:2507.01924*, 2025, doi: 10.48550/arXiv.2507.01924.
- [20] L. Yang, Y. Zhai, Y. Zhang, Y. Zhao, Z. Li, and T. Xu, "A new methodology for anomaly detection of attacks in IEC 61850-based substation system," *J. Inf. Secur. Appl.*, vol. 68, Art. no. 103262, 2022, doi: 10.1016/j.jisa.2022.103262.
- [21] A. Agin, A. Demirenen, O. Usta, and B. Orencik, "Using Deep Learning Algorithms for Cyberattack Detection in IEC 61850-Based Power Systems," in *2024 15th National Conference on Electrical and Electronics Engineering (ELECO)*, pp. 1–6, 2024, doi: 10.1109/ELECO64362.2024.10847104.
- [22] S. Yoon, R. Kim, H. Yoo, and H. Lim, "Real-Time GOOSE Attack Detection in IEC 61850 Substations Using SDN-Based Traffic Inspection," in *Proceedings - 2025 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume, DSN-S*, 2025, pp. 271–272, doi: 10.1109/DSN-S65789.2025.00077.
- [23] Y. Zhong and A. Lisitsa, "Can process mining help in anomaly-based intrusion detection?," *arXiv preprint arXiv:2206.10379*, 2022, doi: 10.48550/arXiv.2206.10379.
- [24] D. Lozano-Paredes, L. Bote-Curiel, J. R. Feijóo-Martínez, I. Gómez-Talal, and J. L. Rojo-Álvarez, "Explainable Autoencoder-Based Anomaly Detection in IEC 61850 GOOSE Networks," pp. 1–13, 2026, [Online]. Available: <http://arxiv.org/abs/2601.09287>.
- [25] H. Zhao, A. Cheng, Y. Wang, S. Wang, and H. Wang, "CAN Intrusion Detection System Based on Data Augmentation and Improved Bi-LSTM," in *Proceedings of the 2024 IEEE 20th Asia Pacific Conference on Circuits and Systems (APCCAS) and IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia)*, 2024, pp. 198–202, doi: 10.1109/APCCAS62602.2024.10808253.
- [26] P. P. Biswas, H. C. Tan, Q. Zhu, Y. Li, D. Mashima, and B. Chen, "A synthesized dataset for cybersecurity study of IEC 61850 based substation," in *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2019, doi: 10.1109/SmartGridComm.2019.8909783.
- [27] Tasmi *et al.*, "Introduction to GOOSE Data Communication Attack Traffic Pattern in IEC 61850," in *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pp. 256–261, 2024, doi: 10.1109/EECSI63442.2024.10776142.
- [28] V. Agate, A. De Paola, P. Ferraro, and G. Lo Re, "MIDES: A multi-layer Intrusion Detection System using ensemble machine learning," *Int. J. Intell. Networks*, vol. 6, no. July, pp. 204–223, 2025, doi: 10.1016/j.ijin.2025.09.001.
- [29] A. Abdullah Alfrhan, R. Hamad Alhusain, and

- R. Ulah Khan, "SMOTE: Class Imbalance Problem in Intrusion Detection System," in *2020 International Conference on Computing and Information Technology, ICCIT 2020*, 2020, pp. 1–5, doi: 10.1109/ICCIT-144147971.2020.9213728.
- [30] Q. H. Nguyen *et al.*, "Influence of data splitting on performance of machine learning models in prediction of shear strength of soil," *Math. Probl. Eng.*, vol. 2021, pp. 1-15 doi: 10.1155/2021/4832864.
- [31] F. I. Komputer and U. A. Purwokerto, "Comparative Study of LSTM and GRU Accuracy in Predicting BBRI Stock Closing Price," *Journal of Applied Informatics and Computing*, vol. 10, no. 1, pp. 837–846, 2026, doi: 10.30871/jaic.v10i1.11938.
- [32] A. Senel and M. Al-Hubaishi, "Advanced Feature Engineering and Machine Learning for Simultaneous Cyber Attack Detection in Smart Digital Substations," in *Proceedings of the 4th International Conference on Intelligent Data Communication Technologies and Internet of Things, IDCIoT 2026*, 2026, pp. 147–153, doi: 10.1109/IDCIoT67589.2026.11455674.
- [33] T. Fernando, G. Ramachandran, M. Vilathgamuwa, and D. Jayalath, "Anomaly Detection for GOOSE Spoofing Attacks in Digital Substations Using Deep Learning Models: A DNN and LSTM Approach," *IEEE Access*, vol. 13, pp. 129709–129720, 2025, doi: 10.1109/ACCESS.2025.3591695.
- [34] J. A. de Oliveira, A. F. P. dos Santos, and R. M. Salles, "RNN for intrusion detection in digital substations based on the IEC 61850," *J. Inf. Secur. Appl.*, vol. 94, p. 104197, 2025, doi: 10.1016/j.jisa.2025.104197.