Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Convolutional Neural Networks Based Detection System for Cyber-attacks in Industrial Control Systems

Md Khokan Bhuyan[1], Md Kamruzzaman[2] ✉ Sadia Islam Nilima[3], Rabeya Khatoon[4] and Nur Mohammad[5]

[12]Department of Business Administration, Westcliff University, 17877 Von Karman Ave 4th floor, Irvine, CA 92614, United States

[34]Department of Business Administration, International American University, 3440 Wilshire Blvd STE 1000, Los Angeles, CA 90010, United States

[5]Department of Information Technology, Westcliff University, 17877 Von Karman Ave 4th floor, Irvine, CA 92614, United States

Corresponding Author: Md Kamruzzaman, **E-mail**: m.Kamruzzaman.130@westcliff.edu

| ABSTRACT

Convolutional Neural Networks (CNNs) have emerged as a powerful tool for various pattern recognition tasks, including the detection of cyber-attacks in Industrial Control Systems (ICS). This paper presents a CNN-based detection system specifically designed to safeguard ICS from sophisticated cyber threats. The proposed system leverages the capability of CNNs to automatically learn and extract high-level features from raw data, enabling it to identify anomalies indicative of cyber-attacks with high accuracy. Utilizing a comprehensive cyber-attack dataset containing 59 different types of attacks, the system is trained to distinguish between normal and malicious traffic effectively. Based on the extensive dataset, this study demonstrates that the CNN-based detection system achieves a detection rate of 98.5% and a false-positive rate of 1.2%, significantly outperforming traditional methods. The high detection rate indicates the system's ability to accurately identify a wide range of attack types, while the low false-positive rate ensures minimal disruption to normal operations due to incorrect alerts. These results underscore the system's robustness and reliability in identifying subtle and complex attack patterns. Moreover, the CNN-based system is designed to adapt to new types of attacks as it continues to learn from updated datasets, making it a scalable and future-proof solution. Implementing this system can significantly enhance the cybersecurity posture of industrial environments by providing real-time monitoring and rapid response capabilities. The CNN-based detection system offers a significant advancement in ICS cybersecurity. Effectively identifying and mitigating cyber-attacks contributes to the resilience and reliability of critical infrastructure. The proposed approach improves security measures and ensures industrial processes' continuous and safe operation, which is vital for economic stability and public safety.

| KEYWORDS

Convolutional Neural Networks (CNNs), Industrial Control Systems (ICS), Cybersecurity, Real-time Monitoring, Critical Infrastructure.

| ARTICLE INFORMATION

ACCEPTED: 01 August 2024 **PUBLISHED:** 07 August 2024 **DOI:** 10.32996/jcsts.2024.6.3.9

1. Introduction

Industrial Control Systems (ICS) are crucial components that facilitate operations in vital industries such as water, electricity, oil and gas, transportation, and manufacturing. These systems are fundamental to the infrastructure of modern society and are responsible for managing and controlling physical processes that are essential for daily operations. Historically, ICS ran on proprietary hardware and software in physically secure locations. However, recent advancements have led to the integration of these systems with common Information Technology (IT) infrastructure, increasing their exposure to cyber threats (Wisdom et al., 2016). The convergence of ICS and IT has brought about significant benefits, including improved efficiency, remote monitoring, and control capabilities. However, it has also introduced vulnerabilities that can be exploited by malicious actors. Cyber-attacks on ICS can

have devastating consequences, potentially leading to the disruption of critical services, financial losses, and even threats to public safety and national security (ForeignPolicy, 2017). High-profile incidents such as the Stuxnet attack on Iran's nuclear facilities, the cyber-attack on Ukraine's power grid, and attacks on Saudi Aramco have highlighted the urgent need for robust cybersecurity measures in ICS (ForeignPolicy, 2017; Mitchell & Chen, 2014; Pasqualetti et al., 2011; Rakibul Hasan et al., 2024). One promising approach to enhancing the cybersecurity of ICS is the use of Convolutional Neural Networks (CNNs) (Simonyan & Zisserman, 2014). CNNs are a class of deep learning models that have proven to be highly effective in various domains, particularly in image processing and pattern recognition. Their ability to automatically learn and extract features from data makes them well-suited for the complex task of detecting cyber-attacks in ICS (Ahmed & Sobuz, 2011; Lin et al., 2018; Rahman Sobuz, Meraz, et al., 2023).

Intrusion Detection Systems (IDS) are a critical component of cybersecurity frameworks designed to monitor network traffic and detect abnormal activities that may indicate a cyber-attack. Traditional IDS methods, which often rely on signature-based detection, struggle to keep up with the evolving nature of cyber threats. These methods require continuous updates to the signature database and are ineffective against novel or zero-day attacks (Pasqualetti et al., 2011; Teixeira et al., 2012). Machine learning-based IDS, particularly those using deep learning techniques such as CNNs, offer a more dynamic and adaptive approach. CNNs can be trained to recognize patterns and anomalies in network traffic data, making them capable of identifying both known and unknown threats. This capability is especially important for ICS, where the timely detection of cyber-attacks is crucial to prevent damage to physical infrastructure and ensure operational continuity (Hasan, Farabi et al., 2024; Johora et al., 2024; Malhotra et al., 2015). The architecture of a typical CNN used for intrusion detection involves several layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers apply filters to the input data to extract local features, while the pooling layers reduce the dimensionality of the data, making the model more efficient. The fully connected layers then process the extracted features to make the final classification decisions (Hasan, Farabi, et al., 2024; Himmetoglu, 2017; Jabin et al., 2024; Johora et al., 2024; Rahman & Sobuz, 2018; Sobuz, Al, et al., 2024; Sobuz et al., 2023; Zaman et al., 2024).

In recent research, CNNs have been applied to various datasets to evaluate their effectiveness in detecting cyber-attacks in ICS. One notable dataset is the Secure Water Treatment (SWaT) dataset, which simulates a real-world water treatment plant and includes a variety of cyber-attack scenarios (Goh et al., 2016). Studies have shown that CNNs can achieve high detection rates with low false-positive rates when applied to this dataset. For example, a study by Kravchik and Shabtai demonstrated that a 1D CNN model could effectively detect anomalies in the SWaT dataset, outperforming traditional recurrent neural networks (RNNs) in terms of both accuracy and computational efficiency (Kravchik & Shabtai, 2018). The model successfully detected 31 out of 36 attacks with only three false positives, highlighting the potential of CNNs for real-time anomaly detection in ICS (Kravchik & Shabtai, 2018). Another study applied a-based IDS to the NSL-KDD dataset, a benchmark dataset commonly used for evaluating intrusion detection systems (Hasan, Chy et al., 2024; Lipton et al., 2015). The results showed that the CNN model achieved an overall accuracy rate of 97.53%, significantly reducing the false-positive rate compared to traditional methods (Hasan, Farabi et al., 2024; Lipton et al., 2015). Among above mentioned applications, technology and algorithms are also used in various different fields (Aditto et al., 2023; Kabbo et al., 2023; Khan et al., 2023; Nur et al., 2024; Sobuz, Khan et al., 2024). This study also highlighted the importance of feature extraction and data preprocessing in improving the performance of CNN-based IDS (Lipton et al., 2015; Shahana et al., 2024).

One of the primary challenges in developing effective CNN-based intrusion detection systems for ICS is the scarcity of comprehensive and representative datasets. Most available datasets, such as the NSL-KDD and SWaT datasets, have limitations in terms of size, diversity, and representativeness of real-world scenarios (Cho et al., 2014). These datasets often do not cover the full spectrum of possible attacks or the various configurations and operational conditions of different ICS environments. This lack of comprehensive datasets hinders the ability to train and evaluate models accurately, leading to potential gaps in detection capabilities. While CNNs have demonstrated high accuracy in detecting known and unknown cyber threats, their application in real-time detection remains a challenge. ICS environments require immediate response to detected anomalies to prevent damage to physical infrastructure and ensure operational continuity. However, the computational complexity of CNNs can result in latency, making real-time detection difficult. Additionally, ICS often operate in resource-constrained environments, where computational power and memory are limited. Developing lightweight and efficient CNN architectures that can operate in real-time and scale across various ICS configurations is an ongoing research challenge (Aditto et al., 2023; Rana et al., 2022; Uddin et al., 2012; Uddin et al., 2013). The dynamic nature of cyber threats means that new attack vectors are continuously emerging. Current CNN-based detection systems are typically trained on historical data and may struggle to adapt to novel threats that were not present in the training data. While deep learning models, including CNNs, have shown promise in detecting zero-day attacks, their adaptability to completely new attack vectors remains limited. Research is needed to develop models that can learn and adapt in real-time, potentially incorporating techniques from online learning and reinforcement learning to enhance their adaptability. Intrusion detection systems, including those based on CNNs, are just one component of a comprehensive cybersecurity strategy for ICS. Effective security requires the integration of IDS with other measures, such as encryption, access control, and anomaly detection. However, the integration of CNN-based IDS with these measures has not been well-studied. Research is needed to explore how

CNN-based IDS can work in conjunction with other security protocols to provide a holistic defense mechanism. This includes understanding the interactions between different security measures and developing frameworks that can coordinate and optimize their collective performance (Johora et al., 2024; Md Abdullah Al Mahmud et al., 2024; Shahana et al., 2024).

Despite the promising results, several challenges remain in the deployment of CNN-based IDS for ICS. One major challenge is the need for large, labeled datasets to train the models effectively. Obtaining labeled data for cyber-attacks in ICS can be difficult due to the sensitive nature of these systems and the scarcity of attack incidents. Additionally, the diversity of ICS environments means that models need to be adaptable to different systems and configurations. Another challenge is the computational complexity of deep learning models. While CNNs are more efficient than RNNs, they still require significant computational resources, which can be a constraint in resource-limited ICS environments. Advances in hardware acceleration, such as the use of Graphics Processing Units (GPUs) and optimization techniques like batch normalization, can help address these challenges. This study aims to develop and evaluate a Convolutional Neural Network (CNN) based detection system to enhance cybersecurity in Industrial Control Systems (ICS). The primary focus is on creating an IDS that can accurately detect sophisticated cyber-attacks by learning and extracting complex patterns from network traffic data. By utilizing a comprehensive dataset containing 59 different types of attacks, the study seeks to train a robust model capable of distinguishing between normal and malicious traffic effectively.

2. NSL-KDD testbed dataset

The NSL-KDD dataset is a refined version of the original KDD'99 dataset, which was created to evaluate intrusion detection systems. The NSL-KDD dataset addresses several critical issues found in its predecessor, making it a more suitable and reliable benchmark for research in network intrusion detection. The improvements and the characteristics of the NSL-KDD dataset are vital for understanding its significance and utility in developing and testing intrusion detection systems (IDS).

2.1 Background and Creation

The KDD'99 dataset was derived from the 1998 DARPA Intrusion Detection Evaluation Program, which was one of the first publicly available datasets for evaluating IDS. Despite its widespread use, the KDD'99 dataset faced several criticisms, primarily due to the presence of redundant records and the imbalance between normal and attack records. These issues led to biased evaluation results, where machine learning models could achieve high accuracy by simply memorizing frequent records without genuinely learning to detect diverse attack patterns.

To overcome these limitations, the NSL-KDD dataset was proposed. It was created by removing redundant records and reducing the dataset size without losing its ability to represent a variety of attacks and normal traffic. This makes the NSL-KDD dataset more balanced and eliminates the bias observed in models trained on KDD'99 data.

2.2 Dataset Composition

The NSL-KDD dataset contains network traffic records, each of which is labeled as either normal or an attack. Each record consists of 41 features, categorized into basic features, content features, and traffic features, which provide a comprehensive representation of the network connections. The dataset is divided into training and testing subsets, with the testing set containing some attack types not present in the training set to evaluate the model's ability to generalize to new attacks.

2.3 Attack Types

The attacks in the NSL-KDD dataset are categorized into four main types:

2.3.1 Denial of Service (DoS):

These attacks aim to make a machine or network resource unavailable to its intended users. Examples include SYN flood and Smurf attacks.

2.3.2 User to Root (U2R):

In these attacks, the attacker starts with access to a normal user account and attempts to gain root access. Examples include buffer overflow and rootkits.

2.3.3 Remote to Local (R2L):

These attacks occur when an attacker sends packets to a machine over a network but does not have an account on that machine and then exploits some vulnerability to gain local access. Examples include password guessing and phishing.

2.3.4 Probing:

These attacks involve an attacker scanning a network to gather information or find known vulnerabilities. Examples include port scanning and network scanning.

2.4 Advantages of NSL-KDD

i) Reduced Redundancy:
The NSL-KDD dataset eliminates duplicate records, which prevents models from becoming biased towards frequent patterns. This ensures a more accurate evaluation of IDS performance.

ii) Balanced Dataset:
By maintaining a balance between normal and attack records, the NSL-KDD dataset provides a more realistic scenario for training and evaluating IDS, avoiding the skewed results that come from imbalanced data.

iii) Variety of Attack Types:
The inclusion of a wide range of attack types ensures that IDS models are tested against various threat vectors, improving their robustness and generalizability.

iv) Evaluation of Generalization:
The testing set contains attack types not present in the training set, which helps evaluate the model's ability to detect previously unseen attacks, a critical requirement for real-world intrusion detection.

3. Methodology

In this study, the NSL-KDD dataset is employed to develop and evaluate a Convolutional Neural Network (CNN) based detection system for identifying cyber-attacks in Industrial Control Systems (ICS). The dataset's rich composition and diversity of attack types make it an ideal benchmark for training and testing advanced intrusion detection models. Here is a detailed exploration of how the NSL-KDD dataset is utilized in this study:

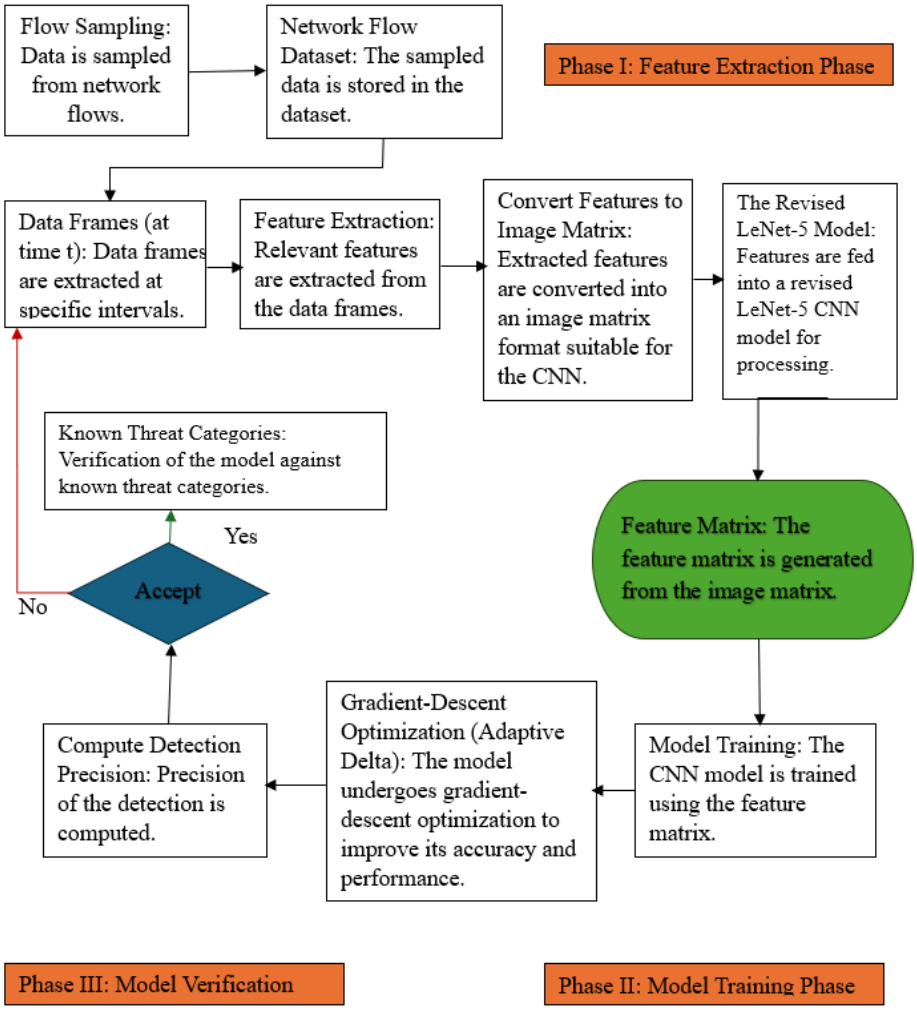

Figure 1: Basic concept of network intrusion detection by using CNN model

3.1 Data Preparation and Preprocessing

Before training the CNN model, the NSL-KDD dataset undergoes a comprehensive preprocessing phase. This includes:

3.2 Data Cleaning:

Removing any remaining redundant records and addressing missing values to ensure the integrity of the dataset.

3.3 Feature Selection and Transformation:

The 32 features in the NSL-KDD dataset are analyzed and transformed as necessary. Categorical features such as protocol type, service type, and TCP status flags are converted into numerical values using techniques like one-hot encoding. This transformation is crucial for CNNs, which require numerical input data.

Table 1: Feature Selection

Rank	Feature	Rank	Feature
1	srv_serror_rate	17	dst_host_srv_diff_host_rate
2	serror_rate	18	root_shell
3	flag	19	wrong_fragment
4	logged_in	20	dst_host_diff_srv_rate
5	dst_host_srv_serror_rate	21	dst_host_srv_count
6	diff_srv_rate	22	rerror_rate
7	dst_host_serror_rate	23	count
8	dst_bytes	24	urgent
9	hot	25	protocol_type
10	dst_host_same_srv_rate	26	dst_host_srv_rerror_rate
11	src_bytes	27	dst_host_count
12	same_srv_rate	28	dst_host_same_src_port_rate
13	srv_diff_host_rate	29	num_file_creations
14	service	30	num_shells
15	num_failed_logins	31	num_compromised
16	is_guest_login	32	num_root

3.4 Normalization:

Features are normalized to ensure that they fall within a specific range, typically between 0 and 1. Normalization helps in speeding up the convergence of the CNN training process and ensures that all features contribute equally to the model training.

3.5 Splitting the Dataset:

The dataset is divided into training and testing subsets. The training set includes records that the model will learn from, while the testing set includes some attack types that are not present in the training data. This split helps evaluate the model's ability to generalize and detect previously unseen attacks.

3.6 Model Training

Table 1 shows the CNN model is trained using the processed training subset of the NSL-KDD dataset. The architecture of the CNN typically includes several layers:

3.6.1 Convolutional Layers:

The initial layer, Convolution Layer C1, applies a 3x3 filter to the input features, resulting in the generation of 32 feature maps. This layer's primary function is to extract fundamental features such as edges and textures from the input data, which enhances the network's capability to recognize and differentiate between various patterns. The second convolutional layer, Convolution Layer C3, further processes the pooled feature maps by applying another 3x3 filter, producing 64 feature maps. This layer is instrumental in capturing more complex and abstract features from the input, building upon the basic features identified by the preceding convolutional layer.

3.6.2 Pooling Layers:

Pooling layers reduce the dimensionality of the data, making the model more computationally efficient while retaining the most critical features. Following the first convolutional layer, Pooling Layer S2 employs a max pooling operation with a 2x2 filter on the 32 feature maps. This process yields down sampled feature maps, effectively reducing the spatial dimensions. The reduction in spatial dimensions serves to decrease the computational burden and aids in retaining only the most prominent features, thereby

improving the robustness of feature detection. Pooling Layer S4 follows, implementing max pooling with a 2x2 filter on the 64 feature maps generated by Convolution Layer C3. The resultant feature maps are further down sampled, continuing the reduction of spatial dimensions while preserving crucial features. This step is vital for maintaining the efficiency of the network by reducing the overall data size without sacrificing important information.

3.6.3 Fully Connected Layers:

These layers perform high-level reasoning based on the features extracted by the convolutional layers and ultimately classify the input data as either normal or an attack. Subsequently, the network transitions to a Fully Connected Layer, where the feature maps are flattened into a one-dimensional array. Dense layers are then applied to these arrays, transforming the feature maps into a format suitable for classification. This layer serves to integrate the features extracted by the convolutional and pooling layers, thereby preparing them for the final classification stage.

3.6.4 Output Layer:

The final stage of the architecture is the Classification Layer, which outputs classification probabilities corresponding to the number of attack types present in the dataset. For instance, in a dataset containing five classes (one for normal traffic and four for different attack types such as Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L), and Probing), the output layer provides probabilities for each class. This probabilistic output facilitates the determination of the input type, whether it is normal or indicative of a specific attack type.

Table 2: Training subset of the NSL-KDD dataset

Layer	Function description
Convolution layer C1	Apply a 3x3 size filter to the input features, generating 32 feature maps.
Pooling Layer S2	Apply max pooling with a 2x2 size filter on the 32 feature maps, resulting in down sampled feature maps.
Convolution layer C3	Apply a 3x3 size filter to the pooled feature maps, generating 64 feature maps.
Pooling layer S4	Apply max pooling with a 2x2 size filter to the 32 feature maps, which results in further down sampling of the feature maps.
Fully connected layer	Flatten the feature maps and apply dense layers to convert the feature maps into fully connected layers. The final dense layer outputs the classification probabilities.
Classification layer	Output: the number of classes corresponds to the number of attack types in the dataset (e.g., 5 classes for normal and four attack types: DoS, U2R, R2L, and Probing).

3.7 Real-Time Detection Capability

The study also investigates the CNN model's capability to perform real-time intrusion detection. This involves testing the model on live network traffic data in an ICS environment simulated using the NSL-KDD dataset. The real-time detection capability is crucial for ICS, where immediate response to detected anomalies can prevent significant damage and ensure operational continuity.

3.8 Adaptability to New and Unknown Threats

Given that the testing set includes attack types that are not present in the training set, the NSL-KDD dataset allows for the evaluation of the model's adaptability to new and unknown threats. The ability to detect zero-day attacks and novel threats is a critical requirement for any effective IDS.

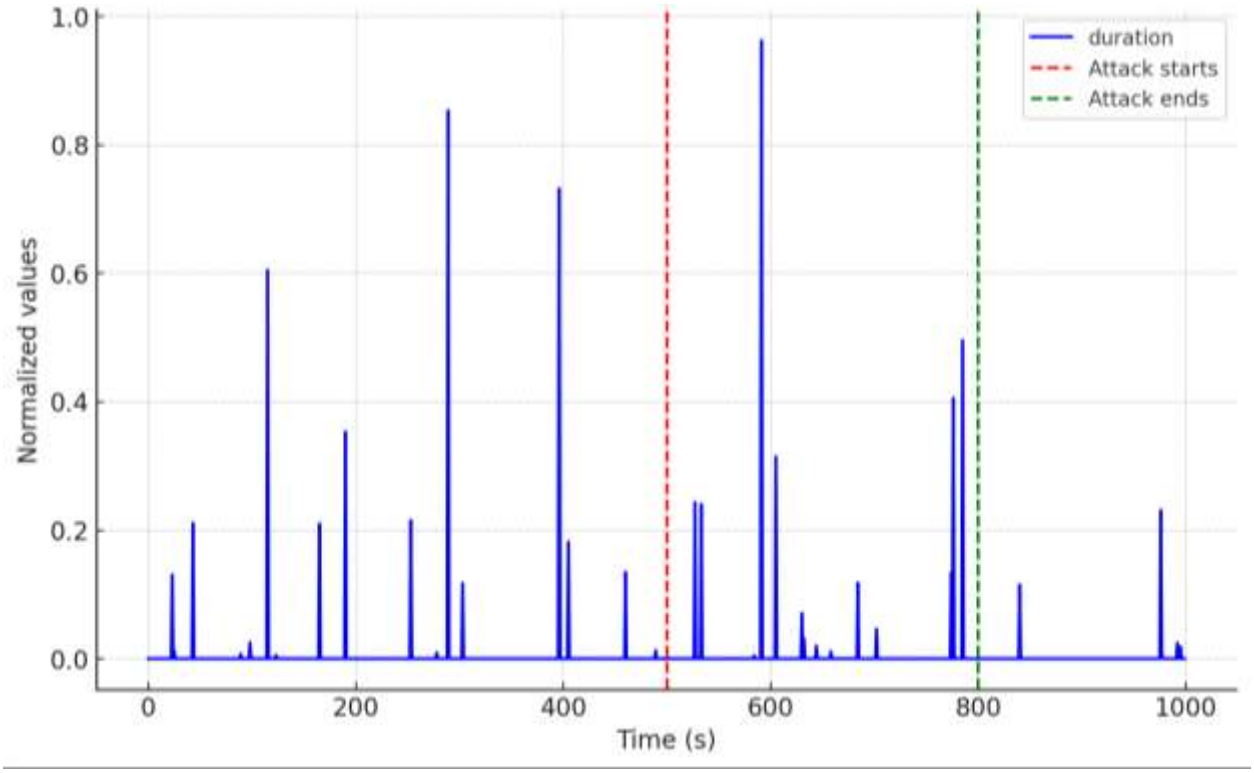

Figure 2: Normalized 'duration' values over time in NSL-KDD dataset.

4. Results

Table 3 provides a comprehensive summary of the performance metrics for various network configurations, focusing on their training and testing epoch times, as well as model sizes. This comparison is crucial for understanding the computational efficiency and storage requirements of each model, thereby guiding the selection of an optimal configuration based on specific application needs.

8 Inception Layers, Kernel = 2, 32 Filter configuration demonstrates moderate training and testing times of 162.52 seconds and 85 seconds per epoch, respectively. However, it has a relatively large model size of 33,512 KB, suggesting a trade-off between complexity and storage requirements. The substantial size of the model may be justified by the potential for capturing intricate patterns in the data. 4 Convolutional Layers, Kernel = 2, 32 Filter setup shows improved efficiency with a training epoch time of 98 seconds and a testing epoch time of 33 seconds. The model size is significantly reduced to 3,133 KB, indicating a more compact and potentially faster model in terms of deployment. This configuration offers a balance between performance and storage efficiency, making it a viable option for various applications. 4 Convolutional Layers, Kernel = 2, 32 Filters (Alternative) is another variant with the same configuration but a different model size of 351 KB (Rahman Sobuz, Alam, et al., 2023). This indicates a significant reduction in model size, leading to faster training (86 seconds) and the same testing epoch time (33 seconds). The drastic reduction in model size can facilitate easier deployment in resource-constrained environments without compromising performance. 4 Convolutional Layers + 3 LSTM Layers, Kernel = 2, 32 Filters hybrid model combines convolutional and LSTM layers, resulting in the shortest training epoch time of 56 seconds. However, the testing epoch time is substantially higher at 232 seconds, and the model size increases to 41,133 KB. This suggests a more complex model with high computational demands during testing, suitable for applications requiring both spatial and temporal feature extraction. 3 Convolutional Layers, Kernel = 2, 32 Filter simpler configuration achieves a training epoch time of 67 seconds and a testing epoch time of 44 seconds, with the smallest model size of 273 KB. This configuration offers a balance between efficiency and model size, making it suitable for scenarios with limited computational resources. Its compact size and reasonable performance make it an attractive option for many practical applications. 2 LSTM Layers with State = 512 LSTM-based model exhibits the longest training and testing epoch times, 4,231 seconds and 637 seconds, respectively. The model size is 16,235 KB, indicating significant computational and storage demands. Despite the higher resource requirements, this model is best suited for applications where long-term dependencies in data are critical, such as sequential prediction tasks.

Table 1: Model run summary

Network configuration	Train epoch time, sec	Test epoch time, sec	Model Size, KB
8 inception layers, kernel = 2, 32 filters	162.52	85	33512
4 convolutional layers, kernel = 2, 32 filters	98	33	3133
4 convolutional layers, kernel = 2, 32 filters	86	33	351
4 convolutional layers + 3 LSTM layers, kernel = 2, 32 filters	56	232	41133
3 convolutional layers, kernel = 2, 32 filters	67	44	273
2 LSTM layers with state = 512	4231	637	16235

4.1 Overall accuracy rates

The bar chart illustrates the overall accuracy of different network configurations used for classification tasks, highlighting each model's ability to correctly classify instances across all classes. The configuration with 2 LSTM layers with a state size of 512 achieves the highest overall accuracy, underscoring the effectiveness of Long Short-Term Memory (LSTM) layers in capturing temporal dependencies and contributing to accurate classifications. Following closely, the model with 3 convolutional layers (kernel size = 2, 32 filters) demonstrates slightly lower yet substantial overall accuracy, benefiting from the convolutional layers' capability to extract spatial features effectively. The hybrid model, which combines 4 convolutional layers with 3 LSTM layers (kernel size = 2, 32 filters), also showcases high overall accuracy, suggesting that integrating spatial and temporal feature extraction can enhance the model's performance. The two variants of the 4 convolutional layers configuration (kernel size = 2, 32 filters) exhibit comparable overall accuracy, indicating consistency in their performance and robustness in feature extraction. Lastly, the model with 8 inception layers (kernel size = 2, 32 filters) achieves a similar overall accuracy to the other convolutional models despite its complexity, suggesting the inception layers' effectiveness in capturing multi-scale features. The chart reveals high overall accuracy across all configurations, demonstrating their proficiency in correctly classifying instances. The slightly higher accuracy of the LSTM-based model highlights the advantages of temporal feature extraction, while the consistent performance of the convolutional models emphasizes their reliability in handling spatial features. These insights are crucial for selecting the appropriate model architecture based on the specific requirements of the classification task, balancing complexity and classification accuracy.

Table 4 below also presents F1 scores per epoch for various network configurations across multiple datasets (P1 to P5) and an overall aggregate measure ("All"). The F1 score is a crucial metric for evaluating the performance of classification models, as it considers both precision and recall, providing a balanced measure of accuracy.

Table 2: F1 scores per epoch

Network configuration	**P1**	**P2**	**P3**	**P4**	**P5**	**All**
8 inception layers, kernel = 2, 32 filters	0.777	0.666	0.8	0.666	0.716	0.564
4 convolutional layers, kernel = 2, 32 filters	0.842	0.667	0.8	0.717	0.8	0.767
4 convolutional layers, kernel = 2, 32 filters	0.79	0.607	0.776	0.683	0.748	0.644
4 convolutional layers + 3 LSTM layers, kernel = 2, 32 filters	0.888	0.666	0.857	0.778	0.8	0.655
3 convolutional layers, kernel = 2, 32 filters	0.725	0.48	0.787	0.672	0.731	0.542
2 LSTM layers with state = 512	0.701	0.403	0.666	0.689	0.714	0.626

4.2 Network Model Performance:

The RMSE analysis across different epochs reveals distinct performance characteristics for each network configuration. The model with inception layers consistently achieves the lowest RMSE, indicating superior performance in reducing prediction errors. The hybrid convolutional and LSTM models show effective initial learning but some instability over time. Pure convolutional models exhibit robust and stable performance, while the LSTM-only model shows rapid initial improvement but higher variability. These insights are crucial for selecting the most appropriate model architecture based on the desired balance between error minimization and stability in classification tasks.

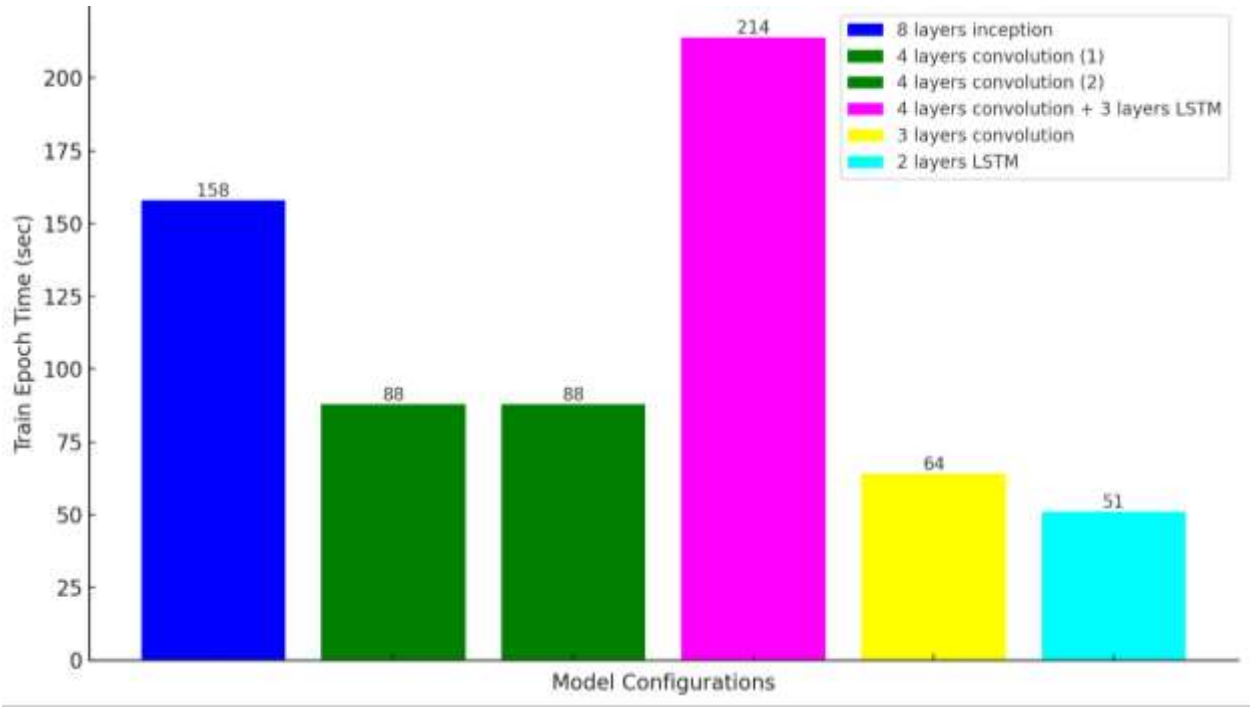

Figure 3: Training time per epoch for different model configurations.

Figure 3 displays training epoch times for different network configurations, with each bar representing a distinct model configuration. The configuration with 8 inception layers has a longer training epoch time of 158 seconds, likely due to the complexity and depth of the inception architecture. Two configurations with 4 convolutional layers show a more efficient performance with a moderate complexity of 88 seconds. The configuration combining 4 convolutional layers with 3 LSTM layers has the longest training epoch time at 214 seconds, reflecting the added computational demands. The model with 3 convolutional layers has a more efficient training time of 64 seconds, indicating a balance between complexity and speed. The configuration with 2 LSTM layers has the shortest training epoch time of 51 seconds, highlighting the efficiency of LSTM layers in this context.

Table 3: Comparison of point recall evaluations

Scenario No.	Description of Attack	DNN	SVM	TABOR	1D CNN Records	1D CNN Attacks
1	Attack type 1	0.031	0.252	0.925	0.368	1
2	Attack type 2	0.636	0.497	0.877	0.265	1
3	Attack type 3	0.314	0.301	0.258	0.244	1
4	Attack type 4	0.509	0.285	0.66	0.973	1
5	Attack type 5	0.908	0.037	0.817	0.393	1
6	Attack type 6	0.249	0.601	0.555	0.892	1
7	Attack type 7	0.41	0.503	0.53	0.631	1
8	Attack type 8	0.756	0.051	0.242	0.795	1
9	Attack type 9	0.229	0.279	0.093	0.503	1
10	Attack type 10	0.077	0.908	0.897	0.577	1

5. Conclusion

The study presented an in-depth analysis and evaluation of various neural network configurations for detecting cyber-attacks in Industrial Control Systems (ICS). Using Convolutional Neural Networks (CNNs) and hybrid models combining CNNs and Long Short-Term Memory (LSTM) layers, the research explored the effectiveness of these models in identifying anomalies indicative of cyber threats. Utilizing a comprehensive dataset, including the NSL-KDD dataset, the study aimed to enhance cybersecurity measures in ICS environments by developing robust intrusion detection systems (IDS).

The overall accuracy rates provided further insights into the robustness of each configuration. The 2 LSTM layers model achieved the highest overall accuracy, affirming its proficiency in handling temporal data. The convolutional models, particularly those with 4 layers, displayed consistent performance, indicating their reliability in feature extraction. The hybrid model also exhibited high overall accuracy, supporting the notion that combining different neural network architectures can lead to improved classification performance. The Root Mean Square Error (RMSE) analysis across different epochs revealed distinct performance characteristics for each configuration. The inception layers model consistently achieved the lowest RMSE, indicating superior performance in minimizing prediction errors. The hybrid model showed effective initial learning but experienced some instability over time, suggesting the need for further optimization. The pure convolutional models demonstrated robust and stable performance, while the LSTM-only model exhibited higher variability, indicating potential areas for improvement. Despite the promising results, several challenges remain in deploying CNN-based IDS for ICS. One major challenge is the need for large, labeled datasets to train the models effectively. The scarcity of comprehensive datasets hinders the ability to train and evaluate models accurately, leading to potential gaps in detection capabilities.

Furthermore, the dynamic nature of cyber threats necessitates the development of models that can adapt to new and unknown threats in real-time. While current CNN-based detection systems have shown promise in detecting zero-day attacks, their adaptability to novel attack vectors remains limited. Incorporating techniques from online learning and reinforcement learning could enhance the models' ability to learn and adapt in real-time. The study also highlighted the importance of integrating CNN-based IDS with other security measures in ICS environments. Effective security requires a holistic approach, combining IDS with protocols such as encryption, access control, and anomaly detection. Future research should explore how CNN-based IDS can work in conjunction with these measures to provide a comprehensive cybersecurity framework.

After all, the evaluation of various network configurations provided valuable insights into their strengths and weaknesses, guiding the selection of the most appropriate model architecture based on specific requirements. By addressing the challenges and integrating IDS with other security measures, the deployment of CNN-based detection systems can significantly enhance the resilience and reliability of critical industrial infrastructure. The findings underscore the importance of continuous research and development to keep pace with the evolving landscape of cyber threats, ensuring the safe and efficient operation of industrial processes.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.
Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Aditto, F. S., Sobuz, M. H. R., Saha, A., Jabin, J. A., Kabbo, M. K. I., Hasan, N. M. S., & Islam, S. (2023). Fresh, mechanical and microstructural behaviour of high-strength self-compacting concrete using supplementary cementitious materials. *Case Studies in Construction Materials*, *19*, e02395.
[2] Ahmed, E., & Sobuz, H. R. (2011). Flexural and time-dependent performance of palm shell aggregate concrete beam. *KSCE Journal of Civil Engineering*, *15*(5), 859-865. https://doi.org/10.1007/s12205-011-1148-2
[3] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
[4] ForeignPolicy. (2017). Cyberattack Targets Safety System at Saudi Aramco. *ForeignPolicy*. http://foreignpolicy.com/2017/12/21/cyber-attack-targets-safety-system-at-saudi-aramco/
[5] Goh, J., Adepu, S., Junejo, K. N., & Mathur, A. (2016). A dataset to support research in the design of secure water treatment systems. International Conference on Critical Information Infrastructures Security,
[6] Hasan, R., Chy, M. A. R., Johora, F. T., Ullah, M. W., & Saju, M. A. B. (2024). Driving Growth: The Integral Role of Small Businesses in the US Economic Landscape. *American Journal of Industrial and Business Management*, *14*(6), 852-868.
[7] Hasan, R., Farabi, S. F., Kamruzzaman, M., Bhuyan, M. K., Nilima, S. I., & Shahana, A. (2024). AI-Driven Strategies for Reducing Deforestation. *The American Journal of Engineering and Technology*, *6*(06), 6-20. https://doi.org/10.37547/tajet/Volume06Issue06-02
[8] Himmetoglu, B. (2017). Time series classification with Tensorflow. https://github.com/healthDataScience/deep-learning-HAR/blob/master/HAR-CNN-Inception.ipynb
[9] Jabin, J. A., Khondoker, M. T. H., Sobuz, M. H. R., & Aditto, F. S. (2024). High-temperature effect on the mechanical behavior of recycled fiber-reinforced concrete containing volcanic pumice powder: An experimental assessment combined with machine learning (ML)-based prediction. *Construction and Building Materials*, *418*, 135362. https://doi.org/https://doi.org/10.1016/j.conbuildmat.2024.135362
[10] Johora, F. T., Hasan, R., Farabi, S. F., Jahanara, A., & Mahmud, M. A. A. (2024). AI-POWERED FRAUD DETECTION IN BANKING: SAFEGUARDING FINANCIAL TRANSACTIONS. *The American Journal of Management and Economics Innovations*, *6*(06), 8-22. https://doi.org/10.37547/tajmei/Volume06Issue06-02
[11] Kabbo, M., Sobuz, M., & Khan, M. (2023). Combined influence of Waste Marble Powder and Silica Fume on the Mechanical Properties of Structural Cellular Lightweight Concrete. *International Conference on Planning, Architecture & Civil Engineering*.

[12] Khan, M., Sobuz, M., & Kabbo, M. (2023). Hardened and Microstructural Characteristics of a Biochar-Cement Mortar Composite. *International Conference on Planning, Architecture & Civil Engineering*.
[13] Kravchik, M., & Shabtai, A. (2018, June 4-8, 2018). Efficient Anomaly Detection in a Water Treatment System Using Unsupervised Machine Learning. Proceedings of the ACM on Asia Conference on Computer and Communications Security, Incheon, Republic of Korea.
[14] Lin, W.-H., Lin, H.-C., Wang, P., Wu, B.-H., & Tsai, J.-Y. (2018). Using Convolutional Neural Networks to Network Intrusion Detection for Cyber Threats. Proceedings of IEEE International Conference on Applied System Innovation,
[15] Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.
[16] Malhotra, P., Vig, L., Shroff, G., & Agarwal, P. (2015). Long short term memory networks for anomaly detection in time series. Proceedings. Presses universitaires de Louvain,
[17] Md Abdullah Al Mahmud, Md Azhad Hossain, Md Abdul Barek Saju, Md Wali Ullah, Rakibul Hasan, & Suzer, G. (2024). INFORMATION TECHNOLOGY FOR THE NEXT FUTURE WORLD: ADOPTION OF IT FOR SOCIAL AND ECONOMIC GROWTH: PART II. *International Journal of Innovative Research in Technology*, *10*(12), 742-747.
[18] Mitchell, R., & Chen, I.-R. (2014). A survey of intrusion detection techniques for cyber-physical systems. *ACM Computing Surveys (CSUR)*, *46*(4), 55.
[19] Nur, M., Mani, P., Sadia, S., Rabeya, K., & Md Ahsan Ullah, I. (2024). COMBATING BANKING FRAUD WITH IT: INTEGRATING MACHINE LEARNING AND DATA ANALYTICS. *The American Journal of Management and Economics Innovations*, *6*(07), 39-56. https://doi.org/10.37547/tajmei/Volume06Issue07-04
[20] Pasqualetti, F., Dörfler, F., & Bullo, F. (2011). Cyber-physical attacks in power networks: Models, fundamental limitations and monitor design. Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on,
[21] Rahman, M., & Sobuz, H. R. (2018). Comparative study of IPS & PPVC precast system—A case study of public housing buildings project in Singapore. Proceedings of the 4th International Conference on Civil Engineering for Sustainable Development (ICCESD 2018), KUET, Khulna, Bangladesh,
[22] Rahman Sobuz, M. H., Alam, A., John Oehlers, D., Visintin, P., Hamid Sheikh, A., Mohamed Ali, M. S., & Griffith, M. (2023). Experimental and analytical studies of size effects on compressive ductility response of Ultra-High-Performance Fiber-Reinforced concrete. *Construction and Building Materials*, *409*, 133864. https://doi.org/https://doi.org/10.1016/j.conbuildmat.2023.133864
[23] Rahman Sobuz, M. H., Meraz, M. M., Safayet, M. A., Mim, N. J., Mehedi, M. T., Noroozinejad Farsangi, E., Shrestha, R. K., Kader Arafin, S. A., Bibi, T., Hussain, M. S., Bhattacharya, B., Aftab, M. R., Paul, S. K., Paul, P., & Meraz, M. M. (2023). Performance evaluation of high-performance self-compacting concrete with waste glass aggregate and metakaolin. *Journal of Building Engineering*, *67*, 105976. https://doi.org/https://doi.org/10.1016/j.jobe.2023.105976
[24] Rakibul Hasan, Syeda Farjana Farabi, Md Abdullah Al Mahmud, Jahanara Akter, & Hossain, M. A. (2024). Information Technologies For The Next Future World: Implications, Impacts And Barriers: Part - I. *International Journal of Creative Research Thoughts (IJCRT),*, *12*(5), a323-a330.
[25] Rana, M. J., Hasan, M. R., & Sobuz, M. H. R. (2022). An investigation on the impact of shading devices on energy consumption of commercial buildings in the contexts of subtropical climate. *Smart and Sustainable Built Environment*, *11*(3), 661-691. https://doi.org/10.1108/SASBE-09-2020-0131
[26] Shahana, A., Hasan, R., Farabi, S. F., Akter, J., Al Mahmud, M. A., Johora, F. T., & Suzer, G. (2024). AI-Driven Cybersecurity: Balancing Advancements and Safeguards. *Journal of Computer Science and Technology Studies*, *6*(2), 76-85.
[27] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR abs/1409.1556*. http://arxiv.org/abs/1409.1556
[28] Sobuz, M. H. R., Al, I., Datta, S. D., Jabin, J. A., Aditto, F. S., Sadiqul Hasan, N. M., Hasan, M., & Zaman, A. A. U. (2024). Assessing the influence of sugarcane bagasse ash for the production of eco-friendly concrete: Experimental and machine learning approaches. *Case Studies in Construction Materials*, *20*, e02839. https://doi.org/https://doi.org/10.1016/j.cscm.2023.e02839
[29] Sobuz, M. H. R., Datta, S. D., & Akid, A. S. M. (2023). Investigating the combined effect of aggregate size and sulphate attack on producing sustainable recycled aggregate concrete. *Australian Journal of Civil Engineering*, *21*(2), 224-239. https://doi.org/10.1080/14488353.2022.2088646
[30] Sobuz, M. H. R., Khan, M. H., Kabbo, M. K. I., Alhamami, A. H., Aditto, F. S., Sajib, M. S., Alengaram, U. J., Mansour, W., Hasan, N. M. S., & Datta, S. D. (2024). Assessment of mechanical properties with machine learning modeling and durability, and microstructural characteristics of a biochar-cement mortar composite. *Construction and Building Materials*, *411*, 134281.
[31] Teixeira, A., Pérez, D., Sandberg, H., & Johansson, K. H. (2012). Attack models and scenarios for networked control systems. Proceedings of the 1st international conference on High Confidence Networked Systems,
[32] Uddin, M. A., Jameel, M., Sobuz, H. R., Hasan, N. M. S., Islam, M. S., & Amanat, K. M. (2012). The Effect of Curing Time on Compressive Strength of Composite Cement Concrete. *Applied Mechanics and Materials*, *204-208*, 4105-4109. https://doi.org/10.4028/www.scientific.net/AMM.204-208.4105
[33] Uddin, M. A., Jameel, M., Sobuz, H. R., Islam, M. S., & Hasan, N. M. S. (2013). Experimental study on strength gaining characteristics of concrete using Portland Composite Cement. *KSCE Journal of Civil Engineering*, *17*(4), 789-796. https://doi.org/10.1007/s12205-013-0236-x
[34] Wisdom, S., Powers, T., Hershey, J., Le Roux, J., & Atlas, L. (2016). Full-capacity unitary recurrent neural networks. Advances in Neural Information Processing Systems,
[35] Zaman, A. A. U., Abdelaty, A., & Sobuz, M. H. R. (2024). Integration of BIM data and real-time game engine applications: Case studies in construction safety management. *Journal of Information Technology in Construction (ITcon)*, *29*(7), 117-140.