Intrusion Detection-Based Ensemble Learning and Microservices for Zero Touch Networks

Neda Bugshan, Ibrahim Khalil, Aditya Pribadi Kalapaaking, and Mohammed Atiquzzaman

The authors propose an ensemble learning-based intrusion detection system for a zero-touch network automation process by leveraging ML and microservice technology to improve the trustworthiness of IIoT systems.

Abstract

The industry has undergone a digital transformation facilitated by the Industrial Internet of Things (IIoT) technology, ushering in the era of Industry 4.0. However, the widespread use of IIoT devices, such as sensors, robots, and other IIoT technologies, has made IIoT systems and associated services vulnerable to a range of network-based attacks, which can impede their performance and disrupt operations. Moreover, the automation process is essential for IoT-based industries to meet future demands. The concept of a zero-touch network has recently evolved to coordinate and manage network resources automatically. Machine learning (ML) plays a critical role in its architecture due to its ability to facilitate a close-loop automation process. With ML, analytical tasks and real-time predictions can be achieved effectively to build smart applications for the early detection of cyberattacks in IIoT systems. The ML components, such as preprocessing, training, and testing, can be subdivided into microservices to improve service while allowing interaction with edge and cloud services. In this article, we propose an ensemble learning-based intrusion detection system (IDS) for a zero-touch network automation process by leveraging ML and microservice technology to improve the trustworthiness of IIoT systems. More specifically, we use the feature selection technique to select significant features and pass them to different models, then blend their predictions based on a stacked ensemble learning approach. Experiments are conducted to evaluate the performance of the proposed framework compared to existing studies.

Introduction

The global transformation of traditional industries to Industry 4.0, driven by the remarkable improvement of key enabling technologies, including the Internet-of-Things (IoT), ML, fast-generation wireless network (5G/6G), and blockchain, has enabled smart industries to improve the quality of their services and increase commercial values. This shapes a promising future in which 5G/6G fuels the IIoT by providing seamless, large-scale, and real-time connections between different entities. Furthermore, numerous communication technologies have emerged due to the recent movement to incremental and scalable industries, concentrating on customizing networks and on-demand manufacturing processes. With networks evolving into increasing software-driven heterogeneous architectures, effective integration, composition, and operation have become significant problems. As a result, correcting network problems and failures has become challenging, increasing configuration and maintenance complexity. To perform automation in such complex systems, a zero-touch network management system has emerged to cope with such problems [1]. With IIoT, collected data can be effectively analyzed by leveraging ML techniques to help management systems process information and optimize, maintain, and manage IIoT devices via a closed-loop approach. Thus, it provides valuable knowledge and makes intelligent decisions possible [2] to solve real-world problems. However, with the high computational resource demands of ML operations, IIoT devices with limited resources cannot provide data warehousing and ML tasks independently [3]. Consequently, cloud computing is the platform chosen by default to provide computing resources and analytical operations based on ML techniques.

However, smart systems based on cloud computing suffer from several issues [3], such as high bandwidth consumption and network latency, which affect the processing time of data analysis services. Besides, traditional services, which are monolithic in nature (the application is designed as one complete unit), have a major drawback: any flaw in its core components impacts the entire system. Furthermore, with the deployment of IIoT-enabled devices in the public domain, industry 4.0 is vulnerable to several new cybersecurity threats where attackers attempt to breach the IIoT networks and launch different attacks, such as distributed denial-of-service (DDoS) attacks. Moreover, they can exploit the vulnerabilities in the system [4] to tap into the IIoT network and inject a malicious package to impair the operation of IIoT-based nodes. Currently, the increased frequency of cyberattacks has become a major concern and one that cannot be avoided. Figure 1 depicts potential issues that could arise in industrial zero-touch networks based on traditional cloud computing.

Edge computing technology has recently gained popularity as a means of tackling several issues mentioned above [3]. It enables different computational operations ranging from storage to service execution to be performed close to consumers and data sources for fast data collection and analysis. Moreover, by integrating microservices, data analytics tasks based on ML techniques can be decomposed into several small, independently-operating services. These services are placed at the edge servers to improve bandwidth usage and provide better distribution of services to ensure the convergence between IIoT and cloud computing.

Furthermore, to protect smart industries from possible network threats and mitigate the risk of

This work is supported by the Australian Research Council Discovery Project (DP210102761)

Digital Object Identifier: 10.1109/MCOM.001.2200535

0163-6804/23/$25.00 © 2023 IEEE

intrusion on IIoT nodes, practical security tools based on ML techniques are needed to filter the continuous flow of data generated by IIoT devices to identify malicious traffic. In this regard, a software solution called the Intrusion Detection System (IDS) is designed to inspect IIoT traffic for indicators of potential intrusions or cyberattacks on IIoT networks. IDS enables tracking of any malevolent behavior or security lapses in the network and alerts the administrator to any possible network breach, then permits precautionary measures to be taken against intrusion threats. Additionally, collaborative learning, such as ensemble learning, can be leveraged to strengthen IDS performance.

An ensemble learning [5] technique is a collaborative ML methodology that improves the final model, giving it better generalization capability and greater accuracy than a single ML model applied to solve different classification problems. It produces better prediction performance by effectively combining the predictions produced by several ML models. There are several types of ensemble learning, such as voting, bagging, boosting, and stacking. The voting ensemble is generally used with a stochastic learning algorithm like neural networks or the same ML models with minor variations in the hyperparameters setting. For a classification problem, there are two types of voting: hard and soft. The main drawback of this approach is that the poor models have a negative effect on the final model. Voting ensembles can be improved by weighted voting and stacking.

In this article, we propose a cybersecurity framework for zero-touch networks driven by IIoT in smart industry applications. We achieve this by exploiting microservice and stacked ensemble learning on an edge and cloud continuum. In the proposed architecture, we develop IDS based on the stacked ensemble learning approach. This allows to combine predictions of state-of-the-art models to strengthen the system's trustworthiness by detecting cyber threats in IIoT networks. The stacking technique was chosen over others, allowing different learning algorithms to be selected as base models. In contrast, with the boosting and bagging approach, the models must be homogeneous. Moreover, there is flexibility in choosing a meta-model that combines the outputs of base models to obtain a model with better performances (e.g., lower bias or variance), whereas bagging and boosting requires a deterministic approach (e.g., averaging process).

The contributions of this work are summarized below:

- An IDS based on ensemble learning for zero-touch networks is proposed using a microservice-based ML framework in Edge-Cloud interaction.
- A stacking-based ensemble learning approach that utilizes different models have been leveraged to improve the trustworthiness of IDS by detecting cyber threats targeting IIoT networks.
- We conduct various experiments utilizing three public datasets to assess the performance of the proposed framework. We compare the final results with those reported in recent studies on IDS to validate the system's reliability regarding evaluation metrics and processing time.

The remaining sections are organized as follows: we summarize relevant studies on IDS in the next section. Then the methodology is described, followed by analyzing the results of the experiments.

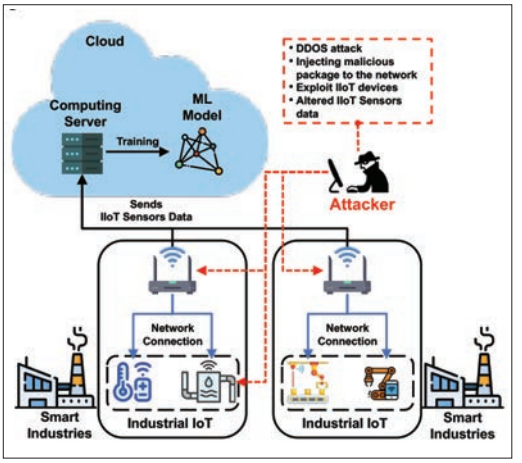

FIGURE 1. Possible threats in industrial zero-touch networks based on conventional cloud computing.

We then discuss future works. Finally, we conclude the article and summarize the main points.

Related Work

A review of previous studies reveals that considerable progress has been made in the development of cutting-edge ML and DL-based algorithms to optimize zero-touch networks and mitigate cybersecurity attacks against the system.

Shaghaghi *et al.* [1] presented a zero-touch network failure recovery paradigm based on reinforcement learning. They used a hybrid neural network (NN) with long short-term memory (LSTM) to detect the time dependency of impending failure. They focus on the optimization problem by minimizing a weighted cost that includes the cost of network resource utilization and the penalty for choosing the wrong option. Similarly, but in a distributed setting utilizing federated learning (FL), Chergui *et al.* [6] proposed an efficient energy management system in the zero-touch network by applying the traditional ML technique. They included a statistical FL-based analytic engine to perform closed-loop automation. Nevertheless, these works do not consider any cyber threats [4] to the IIoT network. Therefore, an advanced IDS is required to enable a zero-touch network to perform efficiently.

Recently, Jayasinghe *et al.* [7] proposed FL-based anomaly detection for zero-touch networks and service management (ZSM). The framework leveraged FL and multi-stage ML models to detect anomalies and explore the possibility of applying them in the ZSM structure. Even though the FL model is applied in multiple stages, this approach is limited to homogeneous models with minor variations in their settings. Furthermore, the poor models have a significant effect on the aggregated model.

About collaborative training based on the ensemble learning concept, Das *et al.* [5] proposed IDS using ensemble learning for feature selection and the training of different ML models. Additionally, they presented a comprehensive review of various ML models and feature selection approaches. To select the best optimal features from the original features set, the ensemble features selection of various methods was applied based on the majority voting technique. However, this approach is computationally expensive as it incorporates several ML models in the first and second-level training

The framework leveraged FL and multi-stage ML models to detect anomalies and explore the possibility of applying them in the ZSM structure. Even though the FL model is applied in multiple stages, this approach is limited to homogeneous models with minor variations in their settings.

A new framework is proposed that leverages microservices to decompose the ML application into smaller and more manageable tasks, thus providing scalability and reusability.

to choose the best-performing ensemble group. Similarly, [8] used the random subspace (RS) technique for random feature selection in combination with a random tree to identify attacks in the network traffic of the SCADA power system. However, the training was performed on separated parts of a dataset without combining them. Moreover, the system's reliability has been measured only in terms of accuracy. In the same vein, [9] used the IIoT-fog-cloud interplay to develop distributed an IDS based on ensemble learning using heterogeneous ML models to detect cyberattacks in the IIoT environment. The feature selection mechanism was based on mutual information. Furthermore, [10] focused on recognizing DDoS attacks against smart contracts in blockchain-based-IIoT networks. They utilized InterPlanetary File System (IPFS) to store incoming traffic from the IIoT layer and fog computing to distribute the training of ML models and to exchange updated parameters via nearby nodes through the primary fog node. Besides, traditional blockchain-based cloud computing stored valid transactions of non-attack traffic classified by the intrusion detection engine. However, storing raw data directly on the blockchain network is impractical due to the high gas consumption cost of conducting a transaction or executing a contract on the Ethereum blockchain platform. This cost is even higher depending on the file size, even for a few kilobytes.

Several studies have taken a DL-based approach. Reference [11] proposed a hybrid framework based on the convolutional neural network (CNN) model for in-depth feature extraction and classical ML models for detecting cyberattacks in SDN networks. Similarly, [12] employed deep transfer learning (DTL) based on the concept of the residual neural network to classify incoming traffic in heterogeneous IIoT networks. However, both approaches suffered from a long training time due to the size of trainable parameters for a deep CNN model. References [13] and [14] developed an IDS framework based on a deep autoencoder (AE) network. In [13], the author ensured data privacy using blockchain and variational AE (VAE). The encoded features produced from VAE passed to the LSTM model to detect anomalies in industrial smart power network traffic. The anomaly detection framework, proposed in [14], comprised a temporal dependencies network powered by LSTM cells to capture long-term dependencies in sequences data and an attention unit to focus on the important information. The isolation Forest is used to detect anomaly points in cyber-physical systems. Nonetheless, the score for precision and F1 were low (71–83 percent) for some benchmark datasets (e.g., ToNIoT).

After reviewing the current studies, it is apparent that some proposed approaches suffer from long training times or low evaluation scores. Therefore, this article focuses on enhancing the reliability of IDS by employing dependable detection mechanisms that result in good evaluation scores based on different metrics and a reasonable training time. A new framework is proposed that leverages microservices to decompose the ML application into smaller and more manageable tasks, thus providing scalability and reusability. Additionally, IPFS and Ethereum blockchains are employed to store hashes of legitimate transactions, resulting in reduced gas consumption.

Proposed Framework

In this section, we present the architecture of the proposed framework and describe its different components. Then, we discuss the learning models involved in the system.

System Overview

The general overview of the proposed framework and the workflow between different components is illustrated in Fig. 2. The proposed framework has three main layers: *Data*, *Edge Intelligence*, and *Cloud Computing*. The description of the layers is presented below.

- ***The Data layer*** is the first layer in our framework. It consists of several IIoT (sensors) devices that monitor specific industrial environments (e.g., energy, agriculture, transportation, etc.) to produce IIoT data presented as $D = \{d_1, d_2, \ldots, d_m\}$. However, this layer and data in transit are prone to various security attacks, such as network-based attacks, which severely affect production and safety and cause disruption of services. Therefore, an intrusion detection engine should be incorporated into the system design to recognize and mitigate security risks in an intelligent manufacturing system. Since IoT devices have limited computing power, this raw data is forwarded to the next layer for further processing and analysis.
- ***The Edge Intelligence layer*** is a core part of our system; here, the collected IIoT data is processed and analyzed using heterogeneous models. This layer facilitates the efficient implementation of a distributed environment in IIoT networks. It consists of the number of local edge servers represented as $E = \{E_1, E_2, \ldots, E_n\}$, which is in the proximity of one or more data sources. Each edge server E_i runs a specific microservice responsible for executing a particular task.

There are four microservices: Data Preprocessing Microservice (DPMS), Model Training Microservice (MTMS), Model Selection Microservice (MSMS), and Model Deployment Microservice (MDMS). The first microservice *DPMS*, applies different preprocessing techniques to the raw input datasets, such as data mapping, to convert non-numerical features to numerical ones and encode string labels with values of zero and one, to indicate normal and abnormal classes, respectively. Additionally, the standardization technique is applied to scale numerical feature values into different scales centered around a mean (μ) of zero and a standard deviation (σ) of one to eliminate any data bias. Furthermore, feature selection is a crucial step that determines the importance of each feature and discards those that are less significant in order to accelerate the processing time and improve the detection rate of intrusion detection models. The next microservice *MTMS*, uses the processed datasets to train different models based on a stacked ensemble learning approach. This approach is discussed in detail in the next section. *MSMS* is the third microservice responsible for selecting the best model version based on the test results of the trained ML models saved in the database and updating the database accordingly. The models are selected according to certain evaluation metrics

(e.g., accuracy). The final microservice is *MDMS*, where the best model is deployed as the final classifier for real-time detection. If the meta-learner has better accuracy than any of the individual learners, then the ensemble version is selected; otherwise, the best individual model is considered.

The deployed detection engine is responsible for classifying incoming IIoT traffic as either normal or abnormal (attack) transactions. The normal IIoT traffic is routed to its destination; subsequently, the IIoT devices can resume normal operations. Otherwise, in case of intrusion (abnormal traffic), the alarm is activated, signaling the need to take appropriate preventative action, such as terminating the activity of malicious devices. Then, the output (transaction is normal or abnormal) is sent to the next layer (cloud computing) for storage purposes. However, data are still susceptible to malicious attacks, including data modification, that can jeopardize smart manufacturing operations. To prevent or mitigate data integrity attacks, IPFS, and blockchain technologies can be used to store only normal traffic transactions. Instead of storing the transaction directly in the blockchain, which negatively impacts the network performance in terms of replication speed through multiple nodes and raises the computation cost of the application, it stores in the IPFS. IPFS has a distributed nature which makes data hacking impossible. Through the smart contract, the resulting hash of the uploaded transaction is stored in the blockchain network hosted in cloud computing.

- ***The cloud computing layer*** is the third layer in the framework. It provides storage and management services and is responsible for storing the details of the transactions (normal and abnormal) that are transmitted from the data intelligence layer. It hosts a blockchain network to store the hashes of valid transactions uploaded to the IPFS in the previous layer. Additionally, the abnormal traffic transactions are simply logged and further analyzed for the global management purposes of IIoT devices: to improve the service offered by the edge intelligence layer.

Within the zero-touch network management system, the different components of layers can collaborate to monitor, collect, and detect malicious traffic in the IIoT network. By leveraging microservices in the proposed framework design, we can decompose the task of ML-based smart IIoT applications into smaller, manageable tasks. These services are distributed at the edge servers to reduce bandwidth usage and latency and to offer better scalability and reusability. Moreover, automation can be achieved by activating the various services mentioned above (*DPMS*, *MTMS*, and *MSMS*) whenever a new training dataset is available and ready; this is done to ensure that the best version of the detection model is selected to be deployed as the final classifier.

Stacked Ensemble Learning-Based Intrusion Detection System (SENS-IDS)

In the proposed framework, we leverage ensemble learning based on the stacking approach to develop an intrusion detection model. The general concept of this approach is that each individual (base) learner is trained on the same training dataset at the first level. In the second level, the outputs (predictions) for the validation and testing set are used as inputs for the meta-learner, which attempts to achieve better results than any single learner by integrating predictions more effectively.

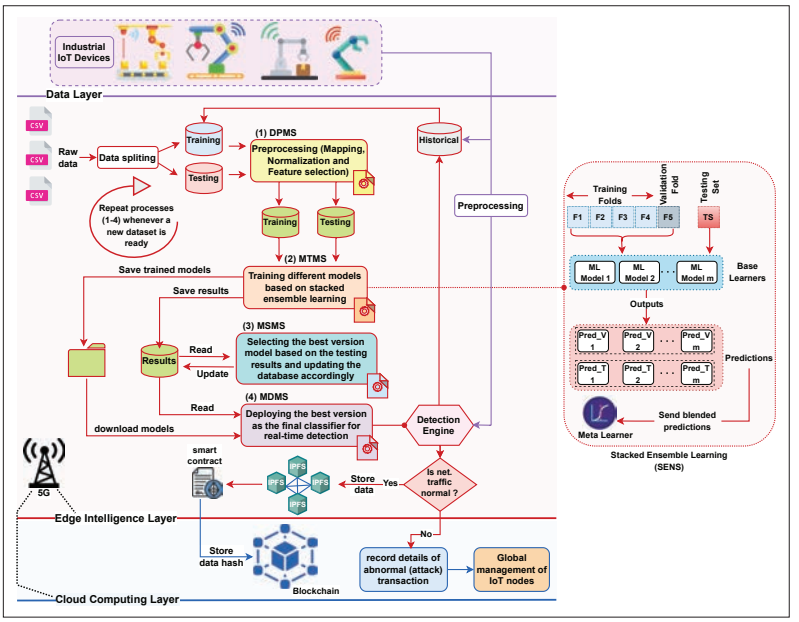

FIGURE 2. Overview of the proposed system: Intrusion Detection based on Ensemble Learning and Microservices for Zero Touch Networks.

We utilize the optimized features from the preprocessing phase to train, validate, and test different models of base learners to produce various outputs (predictions). We use tree-based models, such as Random Forest (RF), Extra Tree (ET), Decision Tree (DT), and Light Gradient Boosting (LGB), due to their effectiveness with complex and nonlinear datasets. Moreover, they can work better with any data type, require minimal data preparation, and perform well even with the default settings. Furthermore, LGB is selected because it is faster than the XGBoost model. Additionally, we use the basic structure of the encoder block of the transformer model [15] with a Multi-Layer Perceptron (MLP) classifier as a head. The transformer is a cutting-edge DL model that uses an attention mechanism to focus on the important parts of input sequences instead of considering all information to find the global interdependence of input and output. It achieves impressive success in natural language processing (NLP) and computer vision (CV) applications. We use five-fold (at each iteration, one fold is used for validation, and the remaining parts are for training) to train various advanced models in the first level. The blended results (predictions of each model) are fed to the Logistic Regression (LR) model to combine these predictions effectively. It is common practice to use a simple linear model as a meta-learner to integrate predictions rather than selecting a complex model such as a Support Vector Machine (SVM) that may entail a longer training time, especially if there are many training samples.

Experiments and Results

This section describes the experimental setting, target datasets, and models and interprets the final results.

Environmental Setup

The experiments were executed on Intel Xeon Silver

The general concept of this approach is that each individual (base) learner is trained on the same training dataset at the first level.

(3.00 GHz) with 4 vCPUs, 12GB RAM, and 1 Nvidia GeForce GTX1080Ti (11GB GPU memory) under Ubuntu 18.04 OS. We reran experiments on Google Collaboratory Pro, which offers an A100-SXM4 GPU with 40GB ram, to test the effect of advanced GPU on training time. We applied the *Synthetic Minority Oversampling* technique, such as SMOTE, to balance the datasets by generating new samples for the minority class. The PermutationImportance and SelectFromModel methods of the sklearn library are utilized for features selection. We used the Sklearn library to implement tree-based models, and the TensorFlow library to implement the DL model. To automate hyperparameter tuning for ML models, we employed the Optuna framework that utilizes Bayesian optimization (by default) and efficient pruning algorithms. For microservice implementation, we used a micro framework called *Flask*. Finally, we used Solidity ver. 0.8.1 to implement a smart contract, Infura IPFS to upload files, and Ganache ver. 2.5.4 for the local testing of the blockchain network.

Dataset and Model

We utilized three public datasets to benchmark the performance of different models. ToNIoT [14] dataset has 19 features and 401,120 ecords, but part of the data is used (86,000 samples). It has been created from diverse sources using several IIoT sensors. The second one has 15 datasets with thousands of samples and 128 features generated from the power system (SCADA) [8] to identify different intrusion attacks. The InSDN dataset [11] has 361,317 samples and 83 features, with 81.06 percent being normal events and 18.94 percent being attack events.

For ToNIoT and InSDN datasets, we used tree-based models, such as RF, ET, and DT, as base models, whereas, for the SCADA dataset, we used RF, ET, and LGB. Each model has different combinations of hyperparameters to experiment with them. For example, in RF and ET models, the number of trees (n_estimators) was set to 100, and the entropy function (criterion) was used to determine the quality of a split. Furthermore, 97 and 47 were utilized as the max depth tree for the DT model for ToNIoT and InSDN, respectively. The learning_rate of 0.7 (how quickly the model can learn) and the maximum leaves number (num_leaves) of 177 were used to tune the LGB model. For the DL model, we used the encoder block of the transformer NN (where it can be stacked multiple times as needed) with the MLP classifier as a head. The number of attention heads is a hyperparameter, and having multiple heads helps the model learn new patterns. Adopting the global average pooling helps reduce the output of the transformer encoder (TE) block to a vector of features before feeding them to the MLP classifier. In general, with the DL models, common hyperparameters, such as batch_size (the number of training observations to be processed), learning_rate, and dropout (randomly drop neurons throughout training), should be carefully calibrated for fast network convergence. The MLP usually has many hidden layers (mathematical functions), each intended to create a particular output.

Evaluation Metrics

We measured the performance of the proposed system utilizing standard metrics, including accuracy (ACC), precision (PR), recall (RC), F1-score, and area under the curve (AUC). PR indicates the percentage of correct positive predictions from the total predictions, whereas RC measures how successful the model is at selecting the correct samples in the set. F1-score is used to calculate the harmonic mean between PR and RC and try to find a balance between them. Furthermore, the ACC is defined as the proportion of correctly predicted outputs to total input samples, which works well if each class has an equal number of samples. Finally, the AUC metric is used for the binary classification problem. It describes the probability of a classifier ranking a randomly selected positive example higher than a negative example.

Results

We compared our results with those reported in recent studies on IDS (summarized in Table 1 and Table 2). The USMD approach experienced declining performance for all metrics except a high recall of 90.01 percent for the ToNIoT dataset. P-ResNet exhibited a good performance by leveraging the CNN-based model, such as residual neural network-based transfer learning. In contrast, TE-IDS improved slightly in terms of most metrics with a superior AUC score of 96.1 percent. However, SENS-IDS outperforms with higher accuracy, precision, recall, F1-score, and AUC of around 94.01 percent, 93.28 percent, 94.84 percent, 94.05 percent, and 95.6 percent, respectively. Regarding the SCADA dataset, RSRT, VAE-LSTM, and TE-IDS demonstrate a comparable accuracy of around 96 percent, with a slight improvement between 1.82 percent-2.51 percent for SENS-IDS. However, SENS-IDS achieved excellent performance (98 percent) in terms of other metrics.

For the purpose of comparison, Table 2 presents the results for the InSDN datasets. By leveraging the convolutional neural network with an enhanced regularizer technique with 48 features group of the InSDN dataset, CNN-Softmax (SD-Reg) achieved outstanding performance on all metrics with slightly better scores towards our proposed approaches of TE-IDS and SENS-IDS. However, the performance of CNN-Softmax (SD-Reg) performance decreased when fewer features (e.g., 9) were used. Specifically, the AUC metric value decreased to 94.4 percent, and recall and F1-score for a normal class decreased to 88.97 percent and 93.98 percent, respectively. Conversely, when 17 features of the InSDN dataset were used, a single neural network-based transformer model (TE-IDS) achieved better scores on all metrics with an AUC value of 99.92 percent. Moreover, SENS-ID, based on a stacked ensemble of different models, achieves a more stable performance with a score of around 99 percent for all metrics.

Additionally, we evaluated the processing time for the proposed framework. Using one GPU (GTX1080Ti), TE-IDS, takes around 350, 1922, and 1178-1470 seconds for ToNIoT, SCADA, and InSDN (17 and 48 features set) datasets compared to 1801.173, 7798.624, and 4753.70-8726.584 seconds for sequential SENS-IDS. However, when an advanced GPU, such as A100, was used, the training time decreased to 140, 424, 753.92-1005.02 seconds for TE-IDS, and the same trend was evident with the SENS-IDS version, with times of 1078.255, 1971.337, and 3439.424-4273.993 seconds for the same datasets mentioned above. However, other factors, such as the dataset size (samples number × features number) and the models' complexity (shallow or deep), also increased

the training time. In the case of DL models, factors such as batch size, learning rate, and the number of epochs also impact the processing time.

It is clear from the results that training DL models, such as the TE model, is time-consuming and more challenging to find optimal hyperparameters than with classical ML models that need less time and tuning. Although this issue is common to DL models, it can be addressed using multiple high-end GPUs to train DL models faster. However, single GPUs are far more costly than CPUs. Since using advanced GPUs results in a high cost, we combine ML models that can perform efficiently in the CPU and only use one DL model that requires an advanced GPU with a reasonable training time. Thus, we can balance the time and the cost to train and evaluate several models for IDS while maintaining cost and time efficiency. Moreover, the testing or real-time detection phase should be faster and take less time, which is the most important aspect of IDS. Comparing the execution time of our system with the times reported in related studies, P-ResNet and USMD took around 24401.586 and 396.317 seconds during the training procedure with the ToNIoT dataset. However, the processing time for the CNN-Softmax (SD-Reg) approach with the InSDN dataset is not mentioned, but the training times required for other public datasets (UNSW-NB15 and CSE-CIC-IDS2018) was 514.391-3463.094 seconds. Moreover, the VAE-LSTM approach recorded an average time of 80–87 seconds, with parts of the dataset containing only 14k samples.

The test accuracy of the final (ensemble) model compared with the single models used for the training phase is shown in Fig. 3. It was evident from the results that the final model outperformed single models. For example, for the ToNIoT dataset, the final model scored 94.01 percent, whereas the accuracy of other tree-based models was around 93 percent, with the TE model having the least accuracy. A similar trend was evident with the SCADA dataset, and the accuracy gap between the final model and other models ranged from 0.99 percent to 2.6 percent. Moreover, the final model still performed better even with InSDN utilizing different feature groups (48 and 17). However, the gap was negligible, ranging from 0.04 percent to 0.17 percent.

Further, Fig. 4 shows the upload time comparison of the blockchain network using a file size of 8.4KB. In the proposed framework, we utilize the blockchain network based on IPFS to save the hash of uploaded data to the immutable ledger. The direct storage of raw data in the blockchain is costly in terms of gas consumption and requires a longer response time (e.g., uploading/downloading). Thus, utilizing IPFS to save data off-chain and hash on-chain is cheaper and more scalable. For example, to store the hash of a file (8KB) in the blockchain as a string, the estimated cost of ether is 0.0021115, while keeping the hash in logs costs around 0.0004955 of ether, compared to 0.01091796 of ether required to store a file directly in the blockchain.

The experimental findings presented above demonstrate that the proposed model is superior to other techniques in terms of all metrics. Moreover, it performs well with fewer features, with less gap between different metrics, and good overall training compared to the single DL models proposed in [11], and [12]. The results indicate that

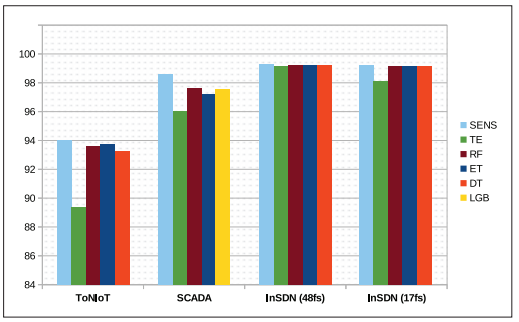

FIGURE 3. The final model (SENS) test accuracy is compared with the average accuracy of each individual model involved in the training phase.

	Technique	ACC	PR	RC	F1	AUC
ToNIoT	USMD [14]	—	70.59	90.01	79.13	70.92
	P-ResNet [12]	87.00	88.00	86.00	86.00	83.00
	TE-IDS	90.12	91.59	87.44	89.46	96.1
	SENS-IDS	94.01	93.28	94.84	94.05	95.6
SCADA	RSRT [8]	96.20	—	—	—	—
	VAE-LSTM [13]	96.27	—	—	—	—
	TE-IDS	96.89	97.78	95.32	96.53	98.6
	SENS-IDS	98.61	98.50	98.55	98.53	98.6

TABLE 1. Results comparison utilizing ToNIoT and SCADA datasets.

the single tree-based models, such as RF, ET, DT, and LGB with default settings, achieve outstanding performance in terms of various evaluation metrics and training time compared to DL models, such as TE, P-ResNe, and CNN-Softmax (SD-Reg). Using all available vCPU cores, the single tree-based model usually needs less training time, not exceeding one minute in most cases. In contrast, DL models are more complex regarding training time and tuning hyperparameters, thereby adding extra complexity to their real-world applications.

Future Research Direction

In this article, we design an IDS based on the stacked ensemble learning technique. However, there remain several challenges requiring further research:

- **Security.** In a zero-touch network, all devices connect to the same network and software. Thus, data leakage may occur during communication, and wirelessly transmitted information is prone to eavesdropping attacks. The application of ML can help to mitigate attacks. However, the ML algorithm provides opportunities for new attacks, such as poisoning and backdoor, that affect the decision-making process. Thus, there is a need to design privacy-preserving models with good performance.
- **Computation complexity.** DL algorithms require much computation to achieve good results. However, a zero-touch network requires efficient computations and instantaneous communication. Therefore, the balance between training time and performance needs further research before applying such models in a real-world scenario.
- **Dataset diversity.** The quality of datasets plays a crucial role in DL performance. Simulating attacks on the network to generate the datasets is insufficient to train a good ML/DL model. Emerging technologies, such as the generative adversarial model (GAN), can be leveraged for further research.

	Technique	ACC (%)	PR (%)		RC (%)		F1 (%)		AUC (%)
			Normal	Attack	Normal	Attack	Normal	Attack	
48fs	CNN-Softmax (SD-Reg) [11]	98.50	99.80	98.27	96.51	98.27	96.51	98.27	98.20
	TE-IDS	99.14	99	99	96	100	97	99	99.96
	SENS-IDS	99.29	99	100	98	100	99	100	99.30
(9,17)fs	CNN-Softmax (SD-Reg) [11]	96.50	99.59	94.72	88.97	99.81	93.98	97.20	94.4
	TE-IDS	98.25	100	97	97	100	98	98	99.82
	SENS-IDS	99.23	99	99	99	99	99	99	99.8

TABLE 2. Results comparison utilizing InSDN dataset.

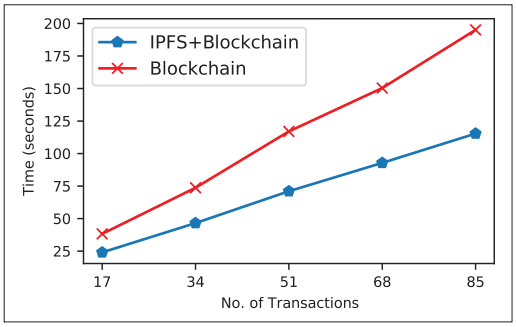

FIGURE 4. Comparison of uploading time using an IPFS+Blockchain vs. Blockchain.

Conclusion

Protecting zero-touch networks in the smart industrial environment from cyberattacks is of utmost importance for long-term success and avoiding catastrophic damage. To ensure the optimal performance of the zero-touch network, an efficient and accurate IDS is necessary. Thus, this article proposes a multi-layered architecture based on the IIoT-edge-cloud interplay to develop an IDS. We used the features selection technique and a stacked ensemble approach to blend the predictions from diverse trained models efficiently. Furthermore, we developed microservices for different ML tasks hosted in the edge layer and incorporated blockchain in the cloud layer. Finally, to assess the system's capability to detect abnormal events in zero-touch networks, we carried out several experiments. The results demonstrate the reliability of the proposed IDS, with a minimal gap between different evaluation metrics and a reasonable execution time compared to existing studies.

References

[1] A. Shaghaghi *et al.*, "Proactive and AOI-Aware Failure Recovery for Stateful NFV-Enabled Zerotouch 6G Networks: Model-Free DRL Approach," *IEEE Trans. Network and Service Management*, vol. 19, no. 1, pp. 437–51, 2022.

[2] M. Friesen, L. Wisniewski, and J. Jasperneite, "Machine Learning for Zero-Touch Management in Heterogeneous Industrial Networks — A Review," *2022 IEEE 18th Int'l. Conf. Factory Communication Systems (WFCS)*, 2022, pp. 1–8.

[3] N. Bugshan *et al.*, "Privacy-Preserving Microservices in Industrial Internet of Things Driven Smart Applications," *IEEE Internet of Things J.*, p. 1–1, 2021.

[4] J. Gallego-Madrid *et al.*, "Machine Learning-Based Zero-Touch Network and Service Management: A survey," *Digital Communications and Networks*, 2021.

[5] S. Das *et al.*, "Network Intrusion Detection and Comparative Analysis Using Ensemble Machine Learning and Feature Selection," *IEEE Trans. Network and Service Management*, 2021, pp. 1–1.

[6] H. Chergui *et al.*, "Zero-Touch AI-Driven Distributed Management for Energy-Efficient 6G Massive Network Slicing," *IEEE Network*, vol. 35, no. 6, 2021, pp. 43–49.

[7] S. Jayasinghe *et al.*, "Federated Learning Based Anomaly Detection as an Enabler for Securing Network and Service Management Automation in Beyond 5G Networks," *2022 Joint European Conf. Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2022, pp. 345–50.

[8] M. M. Hassan *et al.*, "Increasing the Trustworthiness in the industrial IoT Networks Through A Reliable Cyberattack Detection model," *IEEE Trans. Industrial Informatics*, vol. 16, no. 9, 2020, pp. 6154–62.

[9] P. Kumar, G. P. Gupta, and R. Tripathi, "A Distributed Ensemble Design Based Intrusion Detection System Using Fog Computing to Protect the Internet of Things Networks," *J. Ambient Intelligence and Humanized Computing*, vol. 12, no. 10, 2021, pp. 9555–72.

[10] P. Kumar *et al.*, "A Distributed Framework for Detecting DDOS Attacks in Smart Contract-Based Blockchain-IoT Systems by Leveraging Fog Computing," *Trans. Emerging Telecommun. Technologies*, vol. 32, no. 6, 2021, p. e4112.

[11] M. S. ElSayed *et al.*, "A Novel Hybrid Model for Intrusion Detection Systems in SDNS Based on CNN and A New Regularization Technique," *J. Network and Computer Applications*, vol. 191, 2021, p. 103160.

[12] S. T. Mehedi *et al.*, "Dependable Intrusion Detection System for IoT: A Deep Transfer Learning-Based Approach," *IEEE Trans. Industrial Informatics*, 2022.

[13] M. Keshk *et al.*, "A Privacy Preserving-Framework-Based Blockchain and Deep Learning for Protecting Smart Power Networks," *IEEE Trans. Industrial Informatics*, vol. 16, no. 8, 2019, pp. 5110–18.

[14] A. Alsaedi *et al.*, "USMD: Unsupervised Misbehaviour Detection for Multi-Sensor Data," *IEEE Trans. Dependable and Secure Computing*, 2022.

[15] A. Vaswani *et al.*, "Attention is All You Need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

Biographies

Neda Bugshan (neda.bugshan@student.rmit.edu.au) received a master's degree in Computer Science (2012) from RMIT University, Melbourne, VIC, Australia. She is working toward a Ph.D. in computer science from RMIT University. She also works as a Lecturer since 2013 with the Computer Science Department, Applied College — Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. Her research interests include Privacy-Preserving techniques, AI and data analytics security, Distributed Systems, and Natural Language Processing (NLP).

Ibrahim Khalil (ibrahim.khalil@rmit.edu.au) is a Professor in School of Computing Technologies, RMIT University, Melbourne, Australia. He received the Ph.D. degree in Computer Science from the University of Berne, Switzerland, in 2003. He has several years of experience in Silicon Valley Companies. Ibrahim also worked with EPFL and the University of Berne in Switzerland, and Osaka University in Japan. His main research interests are in data privacy and Blockchain. His other research interest include scalable computing in distributed systems, e-health, wireless and body sensor networks, biomedical signal processing, remote health care, network and data security, and secure data analytics.

Aditya Pribadi Kalapaaking (aditya.pribadi.kalapaaking@student.rmit.edu.au) is currently pursuing a Ph.D. degree in Computer Science in School of Computing Technologies, RMIT University, Melbourne, Australia. He received a Bachelor Honours degree in Computer Science from RMIT University, Australia in 2020. His research interests include Machine Learning, Privacy-Preserving techniques, Cybersecurity, Edge Computing, Distributed System, and Blockchain.

Mohammed Atiquzzaman [SM] (atiq@ou.edu) received his M.S. and Ph.D. in electrical engineering and electronics from the University of Manchester, United Kingdom, and B.Sc from Bangladesh University of Engineering and Technology, Bangladesh. He currently holds the Edith Kinney Gaylord Presidential professorship and the Hitachi Chair Professor in the School of Computer Science at the University of Oklahoma. He is the Editor-in-Chief of the *Journal of Networks and Computer Applications*, founding Editor-in-Chief of *Vehicular Communications*, and has served/is serving on the Editorial Boards of various IEEE journals including *IEEE Journal on Selected Areas in Communications*. He co-chaired numerous IEEE international conferences, including IEEE GLOBECOM/ICC. His research interests include communications networks, Internet protocols, wireless and mobile networks, satellite networks, and optical communications.