Neutralising Adversarial Machine Learning in Industrial Control Systems Using Blockchain

Naghmeh Moradpoor*, Masoud Barati†, Andres Robles-Durazno*, Ezra Abah* and James McWhinnie‡
*School of Computing, Edinburgh Napier University, Edinburgh, UK
{n.moradpoor, a.roblesdurazno}@napier.ac.uk, 40482302@live.napier.ac.uk
‡School of Engineering and Built Environment, Edinburgh Napier University, Edinburgh, UK
j.mcwhinnie@napier.ac.uk
†School of Computing, Newcastle University, Newcastle upon Tyne, UK
masoud.barati@newcastle.ac.uk

***Abstract*—The protection of critical national Infrastructures such as drinking water, gas, and electricity is extremely important as nations are dependent on their operation and steadiness. However, despite the value of such utilities their security issues have been poorly addressed which has resulted in a growing number of cyber-attacks with increasing impact and huge consequences. There are many machine learning solutions to detect anomalies against this type of infrastructure given the popularity of such an approach in terms of accuracy and success in detecting zero-day attacks. However, machine learning algorithms are prone to adversarial attacks. In this paper, we propose an energy consumption-based machine learning approach to detect anomalies in a water treatment system and evaluate its robustness against adversarial attacks using our novel dataset. Our evaluations include three popular machine learning algorithms and four categories of adversarial attack set to poison both training and testing data. The captured results show that although some machine learning algorithms are more robust against adversarial confrontations than others, overall, the proposed anomaly detection mechanism which is built on energy consumption metrics and its associated dataset are vulnerable to such attacks. To this end, we propose a blockchain approach to protect the data during the training and testing phases of such machine learning models. We deploy our proposed smart contracts in a public blockchain test network and investigate their costs and mining time.**

***Index Terms*—adversarial attacks, machine learning, critical national infrastructure, industrial control systems, water treatment systems, anomaly detection, blockchain**

I. Introduction

Critical National Infrastructure (CNI), such as: transportation, communication, police systems, national health services, and utilities like: oil, gas, electricity, and drinking water, are a country's public assets. The nation's health and safety and their ability to continue day to day jobs and businesses with no interruption depends on the continuous operation of those assets with no failure and no interruption. However, despite the importance of such assets their cybersecurity issues are poorly addressed. Additionally, the increased level of connectivity for the devices that form a given CNI and the appearance of Industry 4.0 [1] leads to a growing number of cyberattacks against such systems both in occurrence and impact. Criminals and state-sponsored hackers are increasingly going after CNI to disturb society and harm nations.

In 2020, 56% of utility sectors, which include electricity, natural gas, and drinking water, reported at least one cyber-attack on their infrastructure that cause either loss of data or operations shutdown [2]. For example, hackers targeted a U.S. water supply system located in Oldsmar, Florida in 2021 [3] and poisoned the amount of sodium hydroxide, also known as lye, from 100 parts per million to 11,100 parts per million. Luckily the attempt was identified by the operator who successfully reversed the change before the toxic level of chemical reached the drinking water.

Machine learning algorithms have proven their success in detecting known and unknown attacks and producing reliable, repeatable decisions and results in a wide range of networks from traditional computer networks and CNI to wireless technologies. This includes a variety of attacks and applications such as: phishing emails [4], insider threat detection [5], Internet of Things (IoT) attacks [6], mobile malware detection [7], water services [8], and fake news detection [9] as well as predictive maintenance and business process automation.

However, ML techniques are known to be vulnerable to adversarial attacks where hackers and criminals employ the adversarial perturbations during the training and/or testing phases to exploit a given model and cause miss-classification. For example, to classify benign events as malicious, and vice versa, leading to attack detection evasion and disturbance of the systems which force the entire model to fail.

In order to address such adversarial attacks, blockchain-based techniques provide a secure, transparent and immutable way for storing the training or testing data. The third generation of blockchain technology introduces smart contracts enabling combine computer protocols with user interface for executing the conditions/ terms proposed in a real contract. The smart contracts also extend the usability of blockchain-based approaches in various domains or infrastructures (e.g., CNI) so as to record generated critical data in a blockchain network while ensuring their protections under different policies or regulations [22], [23]. The combination of blockchain and machine learning has recently investigated profoundly in order to improve the security of both training and testing data. In [24], a blockchain-based federated learning architecture was presented through which local learning model updates are

securely exchanged and verified using a blockchain network. Moreover, a blockchain-empowered secure data sharing architecture was designed for multiple parties within an industrial IoT environment [25]. The architecture developed a privacy-aware data sharing model using the integration of blockchain and federated learning.

Although the aforementioned blockchain-based approaches have attempted to enhance the security of training/ testing data, none of them had been focused on clean water treatment systems. Additionally, based on our best knowledge, there is no blockchian development nor proposal to protect energy consumption metrics of CNI's endpoints (e.g., sensors and actuators). These features can be employed to detect anomalies against such systems therefore their protection is hugely important. To realise such level of protections, this paper presents the following contributions:

- We implement a virtual testbed representing a clean water treatment system called VNWTS which was designed, implemented, and evaluated during the UK COVID-19 lockdown when accessing our physical testbed was not possible;
- We design a systematical architecture that supports both ML-based engines and a smart contract factory for improving data security against adversarial attacks;
- We define a set of energy consumption features to assist us in detecting anomalies against clean water treatment systems using machine learning algorithms and captured a novel energy-based dataset using various benign & malicious scenarios on the testbed;
- We implement an energy consumption-based machine learning approach to detect anomalies against clean water treatment systems;
- We implement various adversarial attacks, tested our proposed energy consumption-based machine learning approach & its associated dataset against them, and presented the impact of such attacks on the performance;
- We propose a blockchain-based technique to protect our energy consumption-based machine learning approach and its related dataset during the training & testing phases with the aid of smart contracts.

The rest of the paper is structured as follows. Section II designs a blockchain-based and ML-supported architecture for the virtual clean water treatment system and gives the details of its layers. Section III represents the architecture's implementation and describes the interaction among proposed components. Section IV provides some experimental results, and finally Section V concludes the paper.

II. System Architecture

Our proposed system architecture includes six layers of: VNWTS Testbed, Data Management, ML Training Engine, Blockchain Virtual Machine, ML Testing Engine, and Interface as follows.

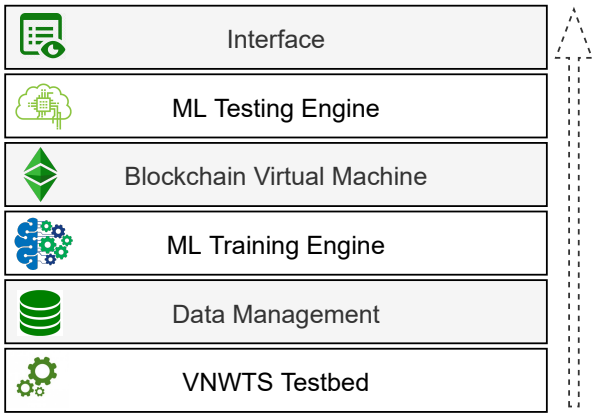

Fig. 1. System architecture

A. Layer 1: VNWTS Testbed

The first layer includes our virtual clean water treatment testbed called VNWTS, which stands for Virtual Napier Water Treatment System, and includes: sensors, actuators, a PLC, a SCADA system, a HMI, and our novel Python code providing communication between the above components. We designed, implemented, and evaluated the VNWTS testbed during the UK COVID-19 lockdown when accessing our physical testbed was not possible because of the strict restrictions. This is comprehensively explained in our previous publications [11-14]. The VNWTS testbed, which is fully explained in the next section, is designed for us to collect a dataset based on energy-based features for anomaly detection in a given clean water treatment system. Another route for us was to go for available data logs such as those from SWaT physical testbed [15]. However, we didn't follow the approaches proposed in [16] - [21], since they didn't include any energy features thus not addressing our needs. Furthermore, our newly collected dataset includes novel attacks on system components such as: level & temperature sensors, hot & cold pump controllers, as well as PLC memory attacks including changing level & temperature setpoints in the working memory which are not present in the existing datasets.

The VNWTS testbed, Figure 2 (left) is implemented in Simulink, a MATLAB-based graphical programming environment, which emulates chlorine treatment of drinking water. Each component of this testbed is a virtual representation of a real element found in the MPA Compact Workstation Rig [10] shown in Figure 2 (right) which represents an excellent scaled-down version of a one of a kind water treatment system. These virtual components have the same characteristics and dynamics of the physical elements from the MPA Compact Workstation Rig and includes: Pipes, Pressure Vessel (x1), Pumps (x2), Proportional Valve (x1), Water Reservoir Tank (x1), Flow Sensors (x2), and Water Supply (x2).

The Pipes used in our virtual model have a 18.621 mm diameter. The Pressure Vessel acts like a normal pipe but because of its different shape it creates a small decline in

water pressure. The Pumps, which include a voltage supplier, a DC Motor, and a centrifuge pump, deliver fluid from the reservoir tanks to another tank (TANK1 in Figure 2, left). The Proportional Valve simulates water demand models for a small city. The Water Reservoir Tank is a virtual representation of the physical tank shown in Figure 2 (right). This tank has the shape of a truncated pyramid.

Flow Sensors allow the rate of fluid to be measured at specific points of our virtual plan and Water Supply, one representing raw water and one representing chlorine.

Additionally, the VNWTS testbed employs a virtual SIMATIC S7-1500 PLC which is available in the SIMATIC S7-PLCSIM Advanced V3.0 software distributed by SIEMENS. Using this software, we successfully emulate the operation of this particular PLC and its internal elements such as Input, Output, working memory, and network functionalities.

Furthermore, we implemented four PI controllers: two to regulate the speed of the pumps delivering the raw water and chlorine, one to regulate the delivery rate for each pump, and one to regulate the water level in the reservoir tank. PI controller is a control mechanism based on mechanical and electronic controllers which consists of two control techniques: proportional and integral.

Moreover, we implemented a Python Communication Module acting as an OPC server allowing the exchange of information between the testbed components. For example, between the PLC and Simulink, where PLC receives the readings from the virtual sensors and controls the actuators such as the pumps discussed above.

B. Layer 2: Data Management

The second layer includes Data Management which contains two components for gathering and handling the project data: Data Collection and Data Pre-Processing.

The Data Collection component gathers the energy traces of the sensors that compose the VNWTS testbed during the simulation run time. The value of each sensor is obtained at a sample rate of 0.1s and saved in a file for later processing. To make the model realistic, we implemented a water demand model of a small city for seven days of a week which is based on a real model of the UK energy consumption. This model has been completely detailed in our previous work [14] and is implemented in a proportional valve of our VNWTS, which is regulated according to the water demand. For example, a fully open valve represents high water demand, while a slightly open valve represents low water consumption. Higher energy consumption is expected during high water demand because the speed of the pumps increases to maintain the level of the reserve tank.

During the benign and attack scenarios, the Data Collection component captures our unique dataset of 3132651 events including eight features such as: Cold Flow Rate, Hot Flow Rate, Temperature, Tank Level, Voltage in the warm water pump, Voltage in the cold water pump, Current in the warm water pump, Current in the cold water pump, along with classification (0 for benign and 1 for attack), and Type of Attack (attack to the level setpoint, attack to the temperature setpoint and attack to multiple sensors).

The Data Pre-Processing component is employed to improve the quality of the raw data previously gathered by the Data Collection module. This phase is extremely important as it has a significant impact on the performance of the machine learning algorithms used in the upper layer. For instance, feature selection could have a huge positive impact in terms of reducing the computational cost of building a predictive model along with improving the performance of it. We have chosen normalization along with three popular feature selection techniques (Information Gain, Chi-Square and Pearson's Correlation) for our data pre-processing phase. The feature selection removed four features thus reduced the total number of features from eight to four: Temperature, Tank Level, Cold Flow Rate, Voltage in cold pump, along with Classification (0 for benign and 1 for attack), and Type of Attack (attack to the level setpoint, attack to the temperature setpoint and attack to multiple sensors).

C. Layer 3: ML Training Engine

The third layer includes a ML Training Engine where the selected machine learning algorithms build models based on sample data, which is also known as "training data", to make prediction or decision (e.g. to predict an event as benign or malicious AKA attack) without being explicitly programmed to do so. For this, we have chosen three popular algorithms: Logistic Regression (LR), Support Vector Machine (SVM), and Artificial Neural Networks (ANN). We used 80% of the collected data for training and 20% for testing. Therefore, in this layer, the ML Training Engine passes the 80% of the pre-processed data to the three ML models built by the above algorithms for the pure purpose of training as the name suggests. Simultaneously, the engine passes the 20% remaining to the next layer, which is the Blockchain Virtual Machine, to store and put aside for the testing phase.

D. Layer 4: Blockchain Virtual Machine

This layer uses a Blockchain-based virtual machine such as Ethereum hosting a smart contract for storing the testing data in a Blockchain. The contract encompasses two functions, called `store()` and `get()`. The former records "Cold Flow Rate, Hot Flow Rate, Temperature, Tank Level, Voltage in the warm water pump, Voltage in the cold water pump, Current in the warm water pump, Current in the cold water pump" in the Blockchain. The *get* function enables users to retrieve the records (testing data) from the Blockchain. The reason of using a public Blockchain here is providing the availability of data with the users in a transparent way.

E. Layer 5: ML Testing Engine

The fifth layer includes a ML Testing Engine where the performance of the three fully trained LR, SVM, and ANN models are evaluated on a testing data which includes the

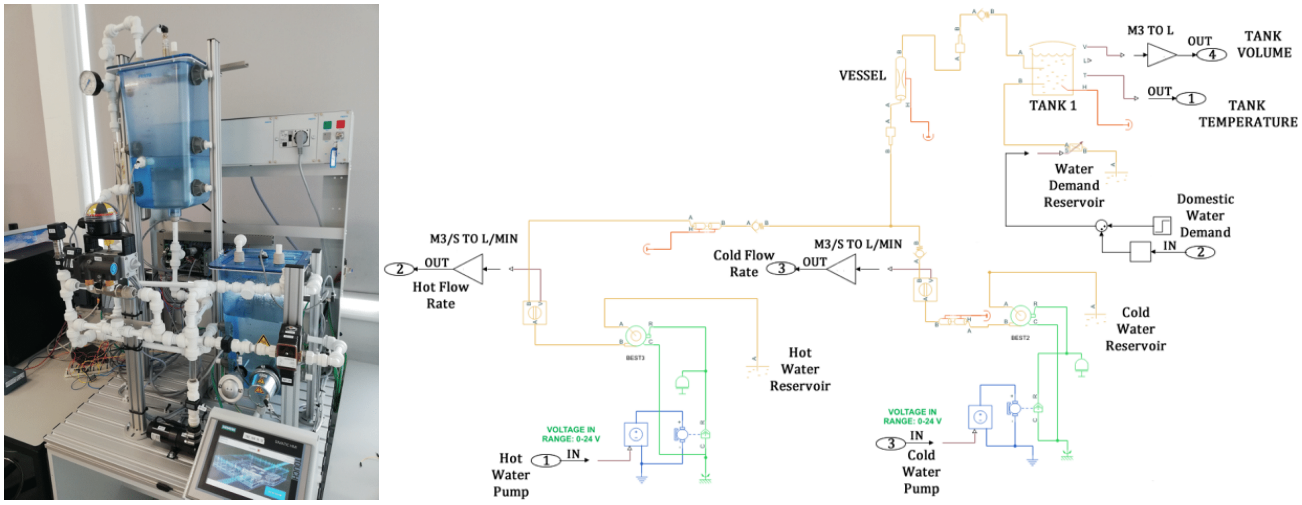

Fig. 2. MPA Compact Workstation Rig & Virtual Water Treatment System

remaining 20% of the total pre-processed dataset. We considered: accuracy, recall, F1 score, and precision as the four main metrics to evaluate the performance of the build models. The ML Testing Engine responsibilities are to: 1) connect to the Blockchain, 2) download the testing data stored previously by the ML Training Engine, 3) pass the testing data to the LR, SVM, and ANN models, and 4) capture the performance of the built models.

F. Layer 6: Interface

This layer enables users to communicate with the system and monitor its functionality as a whole. This includes monitoring that the VNWTS testbed functions correctly (e.g. ensuring that the Python Communication Module allows the exchange of information between the testbed components such as PLC and sensors to control the pumps). It also monitors that the two components of the data management layer, data collection and data pre-processing, work properly. For example, it ensures that: the data collection captures the raw data from the VNWTS testbed taking into consideration the pre-defined energy consumption features, passes them to the data pre-processing component to do normalization and feature selection, and eventually making the data ready for the ML testing and ML training engines. Additionally, the interface layer observes the data split of 80% for the ML training and 20% for the ML testing engines. This is to ensure that the training split is successfully passed to the ML models built by the chosen algorithms, while the testing split is successfully uploaded to the blockchain and downloaded later for testing purpose right after the successful completion of the training phase. The interface is directly connected to a DApp so as to call the *get* function in our proposed smart contract and show the retrieved block contents.

III. Architecture Realisation

The data flow between the different layers of the system is depicted in Figure 3. This includes VNWTS testbed (Layer 1), Data Management (Layer 2), Machine Learning Engine (Layer 3 & 5), and Blockchain Virtual Machine (Layer 4).

The layer one data flow between the system components (Control Station, HMI, PLC, and the Process Under Control also known as Water Treatment System) which forms a SCADA is as follows.

The Control Station loads the program that regulates the water treatment system into the PLC. The Control Station and the PLC communicate over a LAN network. The PLC sends diagnostic information to the Control Station, for example, confirming that the program which regulates the water treatment system is/is not loaded successfully (step 1 in Fig. 3). The Control Station enables the HMI to give direction to the SCADA systems and receive feedback from systems components such as the PLC. The HMI allows a human to control and monitor the water treatment process. The Control Station and the HMI communicate over a LAN network. The HMI sends diagnostic information to the Control Station, for example, confirming that there is/is not a communication issue between itself and the system components (step 2 in Fig. 3). The sensors associated with the system, such as: ultrasonic sensor, flowmeters, and pressure sensor, which are hard-wired to the PLC, provide the status of the water treatment process to the PLC. For example, the ultrasonic sensor provides the water level inside the B102 tank while the flowmeters measure the volumetric flow in the pipes. The PLC implements control techniques such as: PID, Cascade, and Feedforward which manage the actuators such as pumps and valves based on the information received from the hard-wired sensors (step 3 in Fig. 3). The PLC sends information about the water treatment process to the HMI, as a result, line operators can ensure

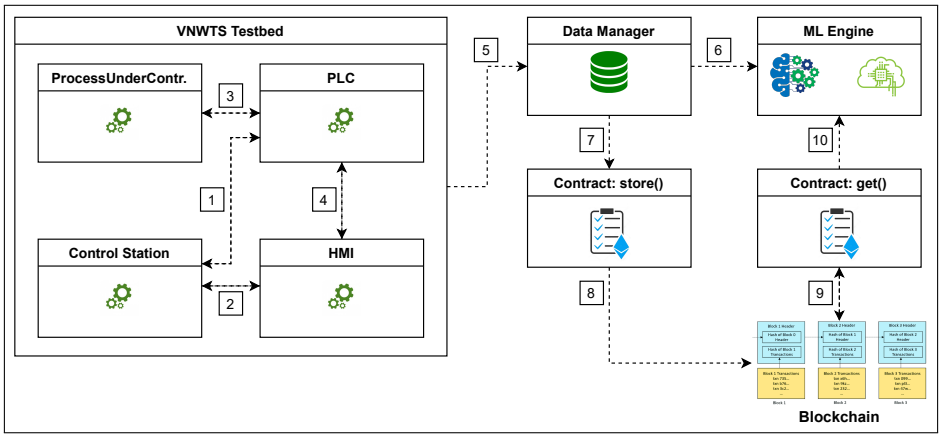

Fig. 3. Interactions within the architecture

that the process is working properly. The HMI is capable of controlling the behaviour of the water treatment process by sending control signals to the actuators or modifying process variables such as setpoints (step 4 in Fig. 3).

Ten system features, which are captured by the ultrasonic and the flowmeter sensors and now form a dataset with millions of both benign and malicious events, pass from the VNWTS testbed to the Data Manager (1) (step 5 in Fig. 3). The features are : Cold Flow Rate, Hot Flow Rate, Temperature, Tank Level, Voltage in the warm water pump, Voltage in the cold water pump, Current in the warm water pump, Current in the cold water pump, and the class feature (0 for benign and 1 for attack), as well as Type of Attack (attack to the level setpoint, attack to the temperature setpoint and attack to multiple sensors). The dataset then goes through a pre-processing phase by the Data Manager. This phase includes: normalization along with three popular feature selection techniques (Information Gain, Chi-Square and Pearson's Correlation). The pre-processed dataset will then split to 80% for training and 20% for testing. The 80% of dataset passes to the ML Engine for creating the machine learning models (using LR, SVM, and ANN ML algorithms) and training them (2) (step 6 in Fig. 3). The remaining 20% will be stored in the blockchain (step 7 in Fig. 3) by deploying the contract and activating *store* function (step 8 in Fig. 3). After building the machine learning models and training them, the final 20% of the dataset, which was previously stored on the blockchain, will be retrieved through the *get* function in the contract and employed to test the ML engine component (steps 9 and 10 in Fig. 3).

IV. Experimental Results

The experiments has two parts:

The blockchain-based evaluation estimates the required gas for the deployment and execution of our proposed smart contracts. Moreover, it investigates the average time taken for the mining process.

A. Adversarial Machine Learning

For the experimental analysis, we implemented a water demand model for a small city inspired by a real model of UK energy consumption for the duration of a week. This includes normal operation and malicious behaviour of the VNWTS testbed. For the malicious scenarios, we developed attacks on VNWTS system components including level & temperature sensors, hot & cold pump controllers, and PLC memory. We categorised the implemented attacks in three groups: attack to the level setpoint, attack to the temperature setpoint, and attack to multiple sensors.

Given the focus of this paper, which is anomaly detection based on energy consumption metrics, and after comprehensive study and our previous research in the field we considered eight energy-based features to capture their values during benign and malicious scenarios. These included: 1) Cold Flow Rate, 2) Hot Flow Rate, 3) Temperature, 4) Tank Level, 5) Voltage in the warm-water pump, 6) Voltage in the cold water pump, 7) Current in the warm water pump, 8) Current in the cold water pump. The dataset also includes binary classification ("0" for benign and "1" for malicious event), and Type of Attack ("1" for attack to the level setpoint, "2" for attack to the temperature setpoint, and "3" for attack to multiple sensors).

We captured 3132651 malicious and benign events which formed a unique energy-based dataset.

We then pre-processed the dataset using normalisation and three popular feature selection techniques (Information Gain, Chi-Square and Pearson's Correlation). While the former reduces data redundancy and improves data integrity, the latter reduces the number of input variables when developing a predictive model. The feature selection techniques, reduced our eight features to four: 1) Temperature, 2) Tank Level, 3) Cold Flow Rate, and 4) Voltage in cold pump. The binary classification ("0" for benign and "1" for malicious event), and Type of Attack ("1" for attack to the level setpoint, "2" for attack to the temperature setpoint, and "3" for attack to multiple sensors) remain unchanged.

We then passed the pre-processed dataset to three popular machine learning algorithms, Logistic Regression (LR), Support Vector Machine (SVM), and Artificial Neural Networks (ANN) to build predictive models equally using 80% of the dataset for training and the remaining 20% for testing.

For the performance metrics of the three algorithms above, we considered: accuracy, recall, f1-score, and precision. However, we only present f1-score and accuracy results in this paper given the page limitation and that f1-score considers recall and precision values in the calculation.

Regarding machine learning adversarial attacks, we consider four categories: random label flipping, targeted label flipping, Fast Gradient Sign Method (FGSM), and Jacobian Saliency Map Attack (JSMA). While flipping techniques (both random and target) focus on training data, FGSM and JSMA targets testing data. We considered these attacks against only one type of classification in our dataset: binary classification ("0" for benign and "1" for malicious event). The impact of adversarial attacks against the other one, which is the multiclass classification where we know the type of attack, will be discusses in our future publications due to lack of space.

We captured the f1-score for the selected ML algorithm after employing random flipping, target flipping, FGSM, and JSMA attacks for the binary classification. They are depicted in Figure 4 – 6. Overall, SVM outperforms LR in terms of showing a longer battle against target flipping attack, however, both algorithms shown the same performance against random flipping, Figure 4. Similarly, ANN shows longer resistance against attacks such as FGSM and JSMA in comparison with LR, Figure 5 -6.

This is the same case for the accuracy reductions for all three algorithms after adversarial attacks. Overall, ANN and SVM reveal a longer resistance against all the attacks (random & target flipping, FGSM, and SJMA) in comparison with LR, Figure 7-9, and target flipping has a greater impact on the accuracy in comparison with random flipping, Figure 7.

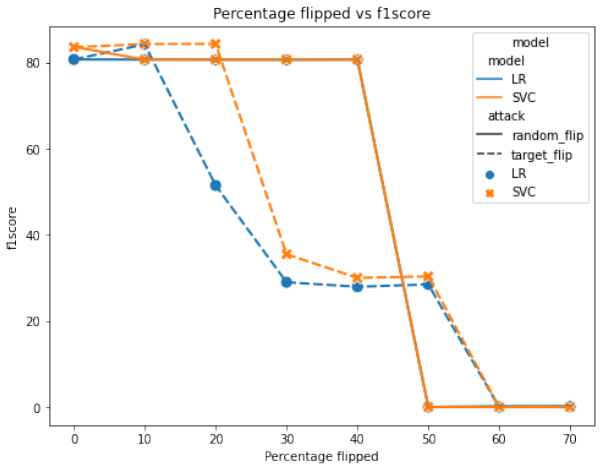

Fig. 4. Comparing f1-score in random & target flipping for LR vs. SVC

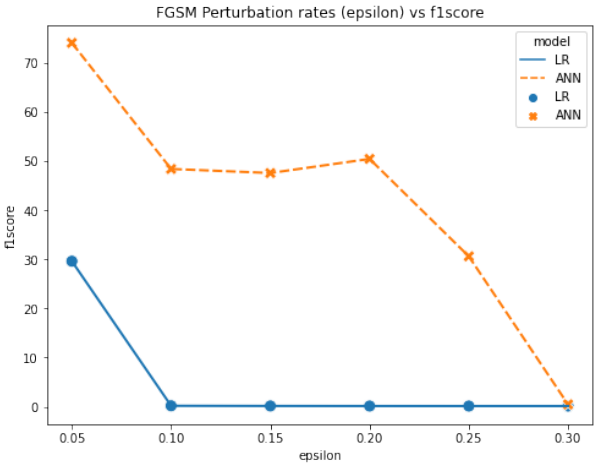

Fig. 5. Comparing f1-score in FGSM for LR vs. ANN

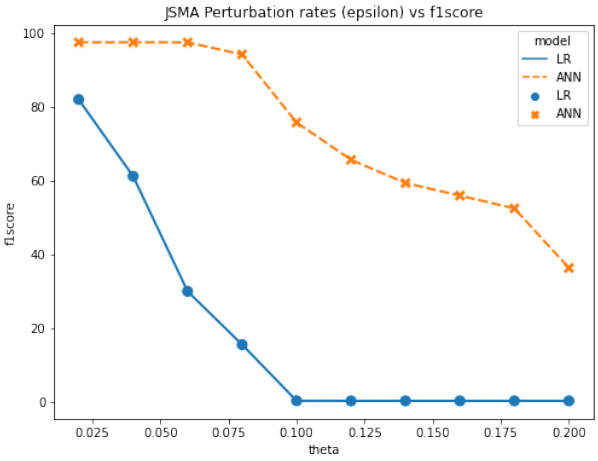

Fig. 6. Comparing f1-score in JSMA for LR vs. ANN

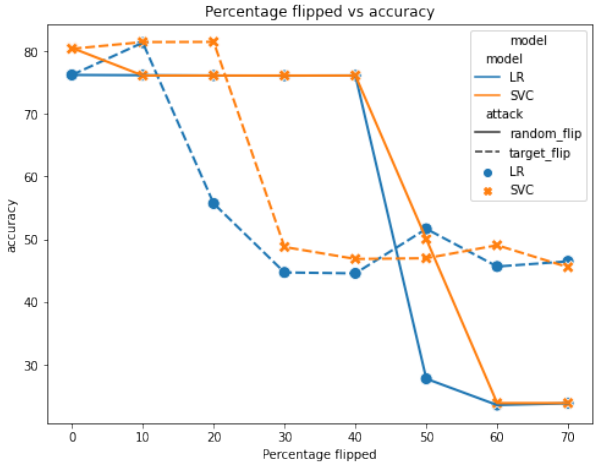

Fig. 7. Comparing accuracy in random & target flipping for LR vs. SVC

B. Blockchain-based Evaluation

We have implemented a Blockchain-based prototype using Ethereum virtual machine and Remix-IDE in order to write

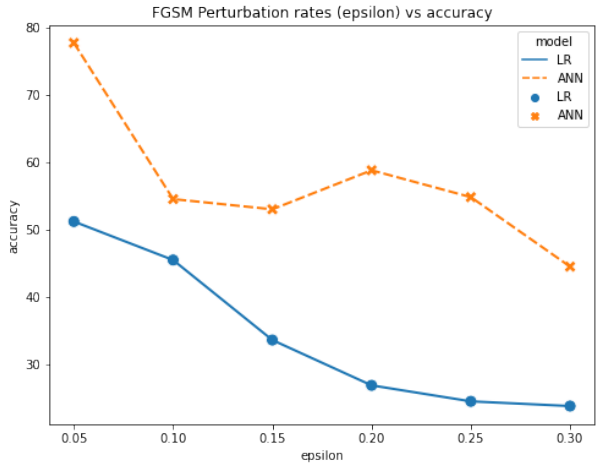

Fig. 8. Comparing accuracy in FGSM for LR vs. ANN

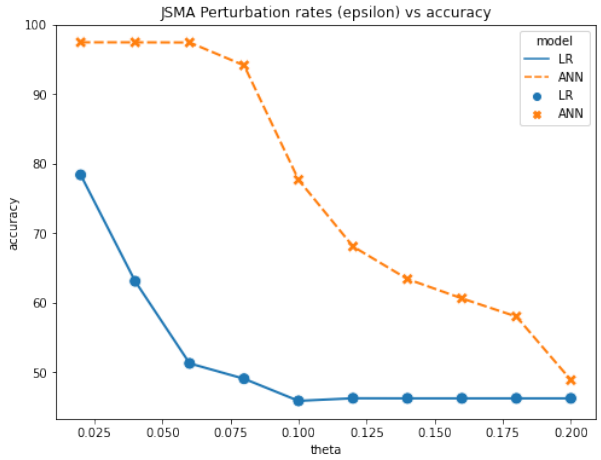

Fig. 9. Comparing accuracy in JSMA for LR vs. ANN

and compile our proposed smart contract [26]. The contract has been written with Solidity, which is a popular programming language for encoding contracts in Ethereum [27]. A public Ethereum test network (Ropsten) was used to deploy our contract and its transactions in a Blockchain network [28]. After the contract deployment, the amount of gas used for its execution was calculated as 244340 *wei*.[1] The average consumed gas was 52700 *wei* for the *store* function and was 35455 *wei* for the *get* function. These results were calculated after five times execution of the functions with different parameters.

Table I represents the average costs and mining time for executing the transactions and creating blocks. The amount of gas prices for cheap, average and fast modes for miners were, respectively, 80, 160 and 320 Gwei. Such values have been captured from ETH Gas station [2] that show the miners' motivation in term of gas price for executing the transactions

[1]Gas is the fee required to successfully run a transaction or deploy a contract on the Ethereum blockchain and its unit is *wei* or *Gwei*.

[2]https://ethgasstation.info/

TABLE I
TRANSACTION COSTS & MINING TIME

Gas Price (Gwei)	80	160	320
Store (Cost: ETH)	0.004	0.008	0.016
Store (Cost: Gwei)	4637600	8432000	16864000
Get (Cost: ETH)	0.003	0.006	0.011
Get (Cost: Gwei)	2836400	5672800	11345600
Mining Time (Sec)	4857	300	28

and blocks creation on the day of our experiment. The average cost for running the *store* function in ETH and Gwei are represented in the table. The cost in Gwei is calculated as: *used gas* $\times$ *gas price*. The same evaluation has been tested for the *get* function. Because the number of opcodes in the *store* function was more than those in the other one, its cost was higher than the *get* function. As seen from the table, when the gas price increase, the average time taken for mining transactions/ blocks reduces sharply. For instance, for a gas price of 320 gwei, miners can create blocks just nearly 28 seconds in average.

V. Conclusion & Future Work

In this paper, we propose an energy consumption-based machine learning approach built on a novel dataset to detect anomalies in a virtual model of a water treatment system named VNWTS. We then evaluate its robustness against adversarial attacks. The evaluation of the proposed anomaly detection algorithm against the adversarial machine learning includes four attack categories: random label flipping, targeted label flipping, Fast Gradient Sign Method, and Jacobian Saliency Map Attack for three popular machine learning algorithms: Support Vector Machine, Logistic Regression, and Deep Learning. Additionally, we consider two popular metrics for performance comparison: f1-score and accuracy. Addressing the captured results, Deep Learning and Support Vector Machine have shown longer battle against all four categories of attack in comparison with Logistic Regression considering both performance metrics. Additionally, the target flipping has a bigger impact compared with random flipping. We conclude that, although there is a different level of resistance among the three algorithms for f1-score and accuracy reduction against adversarial attacks, the proposed energy consumption-based machine learning approach, which is built on the novel energy-base dataset, is vulnerable against such attacks. A smart contract for logging and getting data into/from a blockchain network was deployed in Ropsten and the results showed that an increase in the gas price leads to a noticeable decrease in the average mining time.

Future work will focus on the implementation of the architecture on the real testbed. Moreover, the investigation of our proposed method in a more scalable and decentralised systems using federated machine learning tools and multichain remains another challenge for future direction.

ACKNOWLEDGMENT

This research is supported by the School of Computing and the School of Engineering & the Built Environment at Edinburgh Napier University. The data presented in this study are available on request.

REFERENCES

[1] P. Semwal, "A Multi-Stage Machine Learning Model for Security Analysis in Industrial Control System," In *AI-Enabled Threat Detection and Security Analysis for Industrial IoT*, Springer, Cham, pp. 213–236, 2021.

[2] "Analysis of top 11 cyber attacks on critical infrastructure." Accessed on: Nov. 04, 2021. [Online]. Available: https://www.firstpoint-mg.com/blog/analysis-of-top-11-cyber-attackson-critical-infrastructure/

[3] "U.S. Water Supply System Being Targeted By Cybercriminals." Accessed on: Oct. 18, 2021. [Online]. Available: https://www.forbes.com/sites/jimmagill/2021/07/25/us-water-supply-system-being-targeted-by-cybercriminals/?sh=34c2aa4a28e7

[4] A. Alhogail, and A. Alsabih " Applying machine learning and natural language processing to detect phishing email," In *Computers & Security* , vol. 110, pp. 102414, 2021.

[5] S. Yuan, and X. Wu " Deep learning for insider threat detection: Review, challenges and opportunities," In *Computers & Security* , pp. 102221, 2021.

[6] D. R. Raman, D. Saravanan, R. Parthiban, D. U. Palani, D. D. S. David, S. Usharani, & D. Jayakumar " A Study On Application Of Various Artificial Intelligence Techniques On Internet Of Things," In *European Journal of Molecular & Clinical Medicine*, vol. 7, no. 9, pp. 2531–2557, 2021.

[7] J. M. Arif, M. F. Ab Razak, S. R. T. Mat, S. Awang, N. S. N. Ismail, & A. Firdaus " Android mobile malware detection using fuzzy AHP.," In *Journal of Information Security and Applications* , vol. 61, pp. 102929, 2021.

[8] L. Li, S. Rong, R. Wang, & S. Yu " Recent advances in artificial intelligence and machine learning for nonlinear relationship analysis and process control in drinking water treatment: A review.," In *Chemical Engineering Journal* , vol. 405, pp. 126673, 2021.

[9] R. Jindal, D. Dahiya, D. Sinha, & A. Garg " A Study of Machine Learning Techniques for Fake News Detection and Suggestion of an Ensemble Model.," In *International Conference on Innovative Computing and Communications* , Springer, Singapore, pp. 627–637, 2022.

[10] "MPS PA Filtration Learning System." Accessed on: Oct. 18, 2021. [Online]. Available: https://www.festo-didactic.com/int-en/learning-systems/process-automation/mps-pa-stations-and-complete-systems/mps-pa-filtration-learning-system.htm?fbid=aW50LmVuLjU1Ny4xNy4xOC4xMDgyLjQ3ODU

[11] A. Robles-Durazno, N. Moradpoor, J. McWhinnie, and G. Russell, " Real-time anomaly intrusion detection for a clean water supply system, utilising machine learning with novel energy-based features. " In emph International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 2020.

[12] A. Robles-Durazno, N. Moradpoor, J. McWhinnie, and G. Russell, " A supervised energy monitoring-based machine learning approach for anomaly detection in a clean water supply system. ", In emph IEEE International Conference on Cyber Security and Protection of Digital Services, Glasgow, UK, 2018.

[13] A. Robles-Durazno, N. Moradpoor, J. McWhinnie, G. Russell, and I. Maneru-Marin, " PLC memory attack detection and response in a clean water supply system. ", In emph International Journal of Critical Infrastructure Protections, vol. 26, pp. 100300, 2019.

[14] A. Robles-Durazno, N. Moradpoor, J. McWhinnie, G. Russell, and I. Maneru-Marin, " Implementation and detection of novel attacks to the PLC memory of a clean water supply system. ", In emph International Conference on Technology Trends, pp. 91–103. Springer, Cham, 2018.

[15] A. P. Mathur and N. O. Tippenhauer, " SWaT: A water treatment testbed for research and training on ICS security.", In emph IEEE international workshop on cyber-physical systems for smart water networks (CySWater), pp. 31–36. 2018.

[16] J. Inoue, Y. Yamagata, Y. Chen, C. M. Poskitt, & J. Sun, " Anomaly detection for a water treatment system using unsupervised machine learning.," In *IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 1058–1065, 2017.

[17] J. Goh, S. Adepu, K. N. Junejo, & A. Mathur, " A Dataset to Support Research in the Design of Secure Water Treatment Systems.," In *Critical Information Infrastructures Security* , pp. 88–99, 2017.

[18] J. Goh, S. Adepu, M. Tan, & Z. S. Lee, " Anomaly detection in cyber physical systems using recurrent neural networks.," In *IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, pp. 140–145, 2017.

[19] P. Schneider & K. Böttinger, " High-Performance Unsupervised Anomaly Detection for Cyber-Physical System Networks.," In *Proceedings of the workshop on cyber-physical systems security and privacy* , pp. 1–12, 2018.

[20] K. Yau, K.-P. Chow, & S.-M. Yiu, " Detecting Attacks on a Water Treatment System Using Oneclass Support Vector Machines.," In *IFIP International Conference on Digital Forensics* , Springer, Cham, pp. 95–108, 2020.

[21] A. L. P. Gomez, L. F. Maimo, A. H. Celdran, & F. J. G. Clemente, " MADICS: A methodology for anomaly detection in industrial control systems.," In *Symmetry*, vol. 12, no. 10, pp. 1583, 2020.

[22] B. Faber, G. Michelet, N. Weidmann, R. R. Mukkamala, and R. Vatrapu, "BPDIMS: A Blockchain-based personal data and identity management system", *in Proc. of the 52nd Hawaii International Conference on System Sciences*, Hawaii, USA, 2019, pp. 6855–6864.

[23] M. Barati, O. Rana, I. Petri, and G. Theodorakopoulos, "GDPR compliance verification in Internet of things," *IEEE Access*, vol. 8, pp. 119697–119709, 2020.

[24] H. Kim, J. Park , M. Bennis and S. Kim, Blockchained on-device federated learning, *IEEE Communications Letters*, vol. 24, no. 6, pp. 1279–1283, 2020.

[25] Y. Lu, X. Huang, Y. Dai , S. Maharjan and Y. Zhang, Blockchain and federated learning for privacy-preserved data sharing in industrial IoT, *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2020.

[26] Ethereum, [Online]. Available: https://www.ethereum.org/, Accessed on: Oct 10, 2021.

[27] Solidity, [Online]. Available: https://solidity.readthedocs.io/en/v0.5.3, Accessed on: Oct 10, 2021.

[28] Ropsten testnet pow chain, [Online]. Available: https://github.com/ethereum/ropsten, Accessed on: Oct 10, 2021.