Integrated Smart IoT Infrastructure Management Using Window Blockchain and Whale LSTM Approaches

K. Janani and S. Ramamoorthy

Abstract Internet of Things (IoT) is playing a vital role in the smart infrastructure environment. The IoT vendors are delivering their products in the market without any concern about the security of the devices, so it is an open number of security issues on the IoT devices and data. Security threats are growing high because existing techniques measures are inadequate; two of the most significant concerns in IoT are security and privacy. Due to the IoT devices limited CPU, storage, and energy resources, existing security architectures are unable to provide the key safety requirements, so deep learning and blockchain algorithms are used. These IoT devices give accurate results from heavy complex datasets. Furthermore, blockchain and deep learning model are very familiar to give secured devices to IoT. This proposed model is window blockchain (WBC). In proof-of-work, it leverages past $(n - 1)$ hash to construct the next hash with minimal change; because of this quick block analysis time, we can easily prevent IoT devices from the attackers. WBC's performance is evaluated using an actual data stream generated by one of the analyzed smart infrastructure devices. Another method using deep learning hybrid algorithms for LSTM networks with whale optimization algorithm is a new schema optimization technique that mimics humpback whales' intelligent bubble-net fishing activity. It is an easy, powerful, and predator probabilistic optimization algorithm that can avoid local optima and find global optimal answer. The findings indicate that the proposed window blockchain model improves security and reduces memory utilization this employing limited resources. In the Whale +LSTM (WLSTM), a large number of the dataset were gathered using a real-time scenario using OMNET++IoT plugins, and a Python API is created to insert various malicious activity through networks. The proposed WLSTM model output of 99% has been tested and related to other deep learning utilizing benchmark datasets such as CIDDC-001, UNSWN15, and

K. Janani (✉) · S. Ramamoorthy
Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, India
e-mail: jk6005@srmist.edu.in

S. Ramamoorthy
e-mail: ramamoos@srmist.edu.in

V. Goar et al. (eds.), *Advances in Information Communication Technology and Computing*, Lecture Notes in Networks and Systems 392,
https://doi.org/10.1007/978-981-19-0619-0_10

KDD datasets, as well as actual datasets; the prediction of unknown threats is used to tackle the computation complexity in big networks.

Keywords IoT infrastructure · Security · Window blockchain · LSTM · Whale optimization

1 Introduction

Smart infrastructure is focusing to provide effective solutions. The term "smart city" refers to the use of innovation solutions for improving residents' standard of living, improve government communication, and long term competitive advantage growth. IoT device has increasing count of heterogeneous devices interconnect with Internet [1]. The main challenges are to give safety and security to these devices, which work with lower energy, constraint data, communication protocol, and geographical devices. IoT devices are free to network access, very simple to pull hackers. The combination of these IoT devices with build and workable networks can easily intrude into the IoT network (Fig. 1).

In the blockchain each transactions are stored in blocks of data, which is encrypted by hashing part of the previous $(n - 1)$ block. Blockchain is unchangeable; storage blocks can only be involved. A P2P (peer-peer) network communication for the blockchain of records is permitting requests to access the data carried in every record by communicating the entire model to all the nodes used a blockchain consensus algorithm (BCA) to give the solution to more secure IoT transfer devices. Other than that, it is more secure to apply consensus with hash encryption to give safety interconnected with blocks together [2].

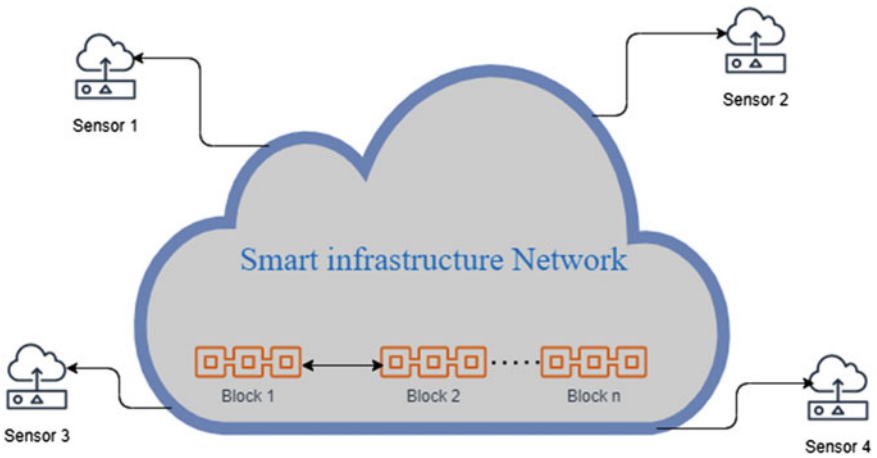

Fig. 1 IoT smart infrastructure network with blockchain security

Block structure in blockchain is a list of blocks that are added to each other in sequential order, almost like a chain. Each block is a data structure containing transactions and is connected to other blocks using an ordered linked list structure. Main data: Transaction data will be stored in blocks. This transaction data is determined by the blockchain's usage factor or the relevant services for which the blockchain is used. Business transaction data would be processed by financial institutions, such as banks. Timestamp: The blocks would also have a timestamp. The timestamp in this case includes the date and time when a specific block is created. Hash: Each block's hash is a unique number created by a cryptography hash algorithm such as SHA-256. The present block's hash as well as the preceding block's hash will be stored in the block. Merkle tree root hash is made up of all the hash for each transaction in a block, and it executes a computational hash calculation to generate a 64-character code. For fast performance and faster data verification, the hashing of the Merkle tree root of all block data is stored. Nonce once is a 4-byte number that is generated at random and can only be used once in an encrypted transaction process. In a proof-of-work algorithm, the nonce is often used as a counter that miners try to solve to generate a new block during the mining process [3] (Fig. 2).

Two of the most significant issues in IoT are security and privacy. Due to the IoT devices' limited CPU, storage, and energy supplies, current security frameworks are still unable to meet the essential security requirements. As a result, the security model for IoT must be distributed and adapted to resource constraints [4]. Blockchain is a decentralized encryption system that can be used for a wide range of purposes. Owing to the high computation time and poor scalability, in its current state, blockchain is unsuitable for IoT. We suggest a blockchain that uses a window block (WBC) that slides through the blockchain. The window starts with a single block and grows to

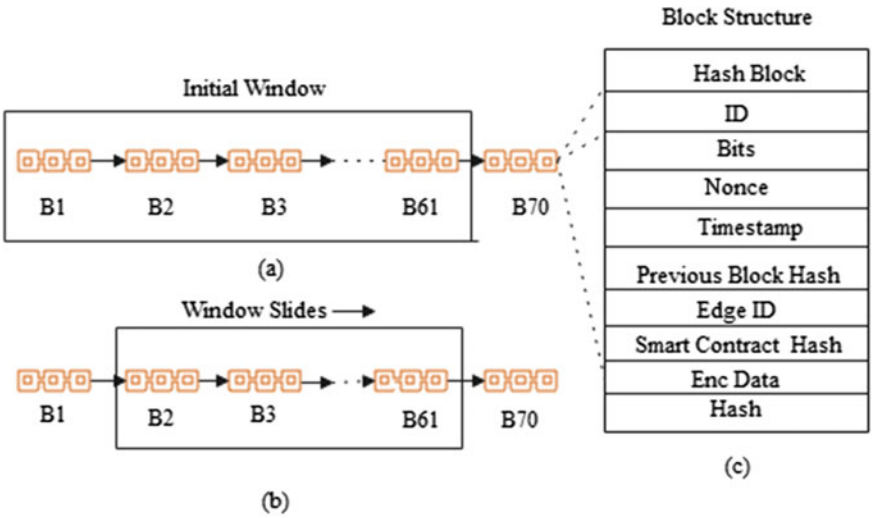

Fig. 2 WBC of window size 61 blocks. (**a**) Initial window. (**b**) WBC. (**c**) Block structure

multiple blocks depending on the hash block size. When making a new block, the blocks in the WBC are used in the architecture that has been proposed; by each hashing the BC inside the window size (WS), the block hash is created. The series of similar previous WBC used to execute and update hash functions is affected by the structure of the window. For a constant complexity of mining, the WBC has a communication complexity of O (n), where n is the count of WBC used in the hash update [5].

Resources like computational capability, energy sources, and memory are all limited in IoT devices. As a result, traditional security algorithms are unsuitable for IoT. A window blockchain that reduces memory overhead and limits communication complexity to meet the needs of a resource-constrained IoT system in the space overhead is reduced by storing only a portion of the blockchain in the IoT unit, as specified by the WS, and keeping the entire blockchain in the private cloud. Using a complexity level around 1 and 5 and omitting the Merkle tree reduces computational workload. The block hash is produced using the attributes of n blocks within the window, which increases protection. The preceding $(n - 1)$ blocks and the window size details are required for a fake miner to mine a block. The window represents a linked list sequence of data that are contiguous or adjacent to one another so the best way can imagine this is maybe an array of characters or an array of integer value and slap of window some subset of this array whatever question we have trying to optimize giving the solution in slid the window over which is find the best part of the solution [6, 7].

Deep learning is an ML process originating from artificial neural network. The network is adjusted to thinking about as variables connected via weighted path. Supervised (SL) or unsupervised (USL) learning group of association inside the network to reach the desired group of output [8]. The learning is carried forward by using labeled and unlabeled sources from supervised or unsupervised deep learning techniques followed by additional compromise of the weights among every pair of neurons. Several IoT-based devices produced a huge number of data DL methods that activate the deep linking of the IoT smart environment [9].

Which is a unified protocol that allows IoT-based application, and device will permit transfer from one to another casually without user intervention. Deep learning LSTM has many benefits; it has many disadvantages when dealing with large datasets. This necessitates the use of a large number of memory cells, which increases computational complexity. This type of problem can also lead to computational complexity. To reduce this issue, you'll need a computationally model for predicting the various types of visible and invisible IoT attacks. To meet the above requirements, a more developed, quick, reliable, and accurate model is needed. The current study aims to establish new hybrid algorithms for whale algorithm in LSTM networks [10].

The first layer of IoT network simulation is built on IP addresses. In both normal and malicious scenarios, a second layer data gathering unit has been added to collect packets. And in the third layer, various attributes were retrieved from the compressed data and utilized to train the proposed DL model to anticipate threats [11]. Finally, the whale-integrated LSTM networks were used to forecast attackers as one of five primary forms of common threats: MIM, DDoS, Dos, data leakage, and spoofing,

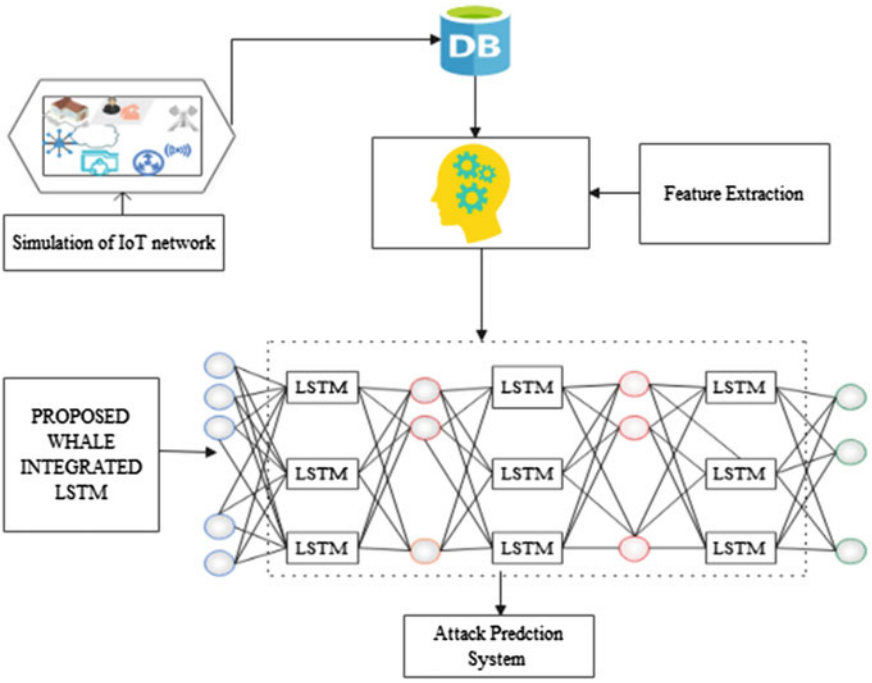

Fig. 3 Proposed model integrated WLSTM

and the system attack forecast as to whether the node is malicious or not, the sort of attack that happened, the Mac address of the devices under assault, and the measures that should be taken [12] Fig. 3.

Contribution of research work as follows:

- The WBC model for the Internet of things is designed to give privacy between the minimum of infrastructure devices.
- On the infrastructure-simulated devices, efficiency and security of the new WBC model were evaluated.
- When using large datasets, reducing the computational complexity needs a more efficient and highly accurate model, so developed the new hybrid algorithm by integrating the whale algorithms in the LSTM network.
- The prediction of unknown threats is used to tackle the computation complexity in large networks.

2 Literature Survey

See Table 1.

Table 1 Survey on smart infrastructure security

Author	Algorithm/protocol	IoT security problem	Implementation	Accuracy	Proposed method	Future work
Luca Barbierato et al.	NB-IoT	Devices that are not capable of actively monitoring and continuously reporting information are considered passive (dump devices)	BC95-B8 module, which operates at 900 MHz	Micro-control extended-309.4 days	Using narrow band IoT, dumb devices may provide real-time or periodic context awareness and data exchange (NB-IoT)	The identification and processing of vast heterogeneous data, data security assurance and user authentication, and data access control are all areas of future study
Petros Spachos et al.	BLE	Improve IoT indoor localization	Kalman filters boost localization, accuracy, and reliability without depending on the cloud	87.5	In a museum, BLE beacons are used to increase interaction. Aim of providing distance and localization facilities, iBeacons have been used	BLE beacons can boost museum interaction at a low cost without any interference with other wireless infrastructure and services in the region
Zhijun Li et al.	BLE2LoRa, CTC	Long distance, long delay, emergency alert	Our BLE2LoRa prototype was built using a USRP B210 (with LoRaWAN PHY) and popular BLE chips (CC1200)	80% Frame over 600 m, 20X	BLE2LoRa is a new CTC solution for direct contact between BLE and LoRa. selecting package bits from a valid BLE frame	Need to build with low cost

(continued)

Table 1 (continued)

Author	Algorithm/protocol	IoT security problem	Implementation	Accuracy	Proposed method	Future work
Zhihan Lv et al.	Zigbee	Humidity, temperature, and battery voltage in smart infrastructure, infrastructure monitor	A simulation validation experiment has been carried out with real-time traffic 20 streetlights counted from A to T as test objects	65RH, 23.5C	The streetlights serve as roads, and the taxis serve as nodes in a simple surveillance system	The different application can try with this network
Dinan Fakhri et al.	MQTT, bitcoin	IoT security problem	Worship, JSON format, Keccak-256, SHA-256	100%	The results of the tests show that an IoT system blockchain-based is more secure than an IoT device that does not use blockchain technology	Future work try with Ethereum
Nabila Islam et al.	Proof of concept	The cyberattack, physical tampering, denial of sleep, DoS for aircraft's data security	NASA's dataset, MATLAB	50 MB take 2500 ms of time	Proofs of concept have been developed to help people recognize how blockchain can function in real-world scenarios	We will improve the accuracy of our research by incorporating the concepts of machine learning and data mining, which will predict whether there will be problems with the planes and eventually reduce time

(continued)

Table 1 (continued)

Author	Algorithm/protocol	IoT security problem	Implementation	Accuracy	Proposed method	Future work
Ali Dorri et al.	Distributed throughput management	Reduces the processing mining delay and overhead	Simulation	51% of attack detected using simulation get high-level accuracy	Complexity, bandwidth and latency overheads, and scalability are all factors to consider. Proposed an LSB for IoT to solve these issues	To gain a better understanding of LSB's output in the real world, create a sample implementation
Safa Otoum et al.	RBC-IDS, ASCH-IDS	Boltzmann ML-based IDS, WSN-based critical infrastructure monitoring	The sensors are spread out over a 100 m × 100 m region in four clusters. The RBM model used for the simulations has one input data (V1) with training data, as well as three hidden layer layers (H1, H2, and H3) DR% = TP/(TP + FP)	RBC-99.12% H = 3–99.91%	When compared to an adaptive machine learning-based IDS solution, the clustered restricted Boltzmann machine-intrusion detection system (RBC-IDS) is an IDS known as the clustered restricted Boltzmann machine-intrusion detection system (RBC-IDS)	Extending the present IDS to wider networks with much more devices is on the future agenda

(continued)

Table 1 (continued)

Author	Algorithm/protocol	IoT security problem	Implementation	Accuracy	Proposed method	Future work
Gonzalo De La Torre Parra et al.	Distributed convolutional neural network cloud-based LSTM	Malicious, phishing, and Botnet attack, DDoS attacks	NB IoT dataset	Phishing attck-94.3%, LSTM-93.58%, Botnet-94.80%	A CNN model is used to detect URL-based attacks on a client's IoT computers. For identifying botnet attacks in IoT devices, the add-on operates in collaboration with an RNNLSTM model hosted on the back-end databases	Includes expanding the proposed technique to critical theory IoT device and machine attacks, such as those that use encrypted traffic to distort or escape detection
Bambang Susilo et al.	CNN, MLP	DoS	Python language, TensorFlow, seaborn	Epoch CNN-90.87%, MLP-54.10%	In an IoT network, different ML and DL algorithms were investigated. The analysis of RF, CNN, and MLP algorithms was included. In terms of effectiveness and AUC for classification tasks, decision trees and CNN got the best results	The NIDS is supposed to incorporate this algorithm

(continued)

Table 1 (continued)

Author	Algorithm/protocol	IoT security problem	Implementation	Accuracy	Proposed method	Future work
Feng Jiang et al.	LSTM-RNN, SVM	Multichannel intelligent attack	DR = TP|(TP + FN), FAR = FP|(TN + FP)	GRNN—87.54%, PNN—96.66%, RBNN—93.05%, KNN—90.74%, SVM—90.4% Bayesian—88.46%, proposed—98.94%	LSTM-RNN is used in multi-channel storage to create classifiers that are used to distinguish the attack from normal traffic to maintain the attack feature of input traffic data	Other researchers would be inspired to build effective deep neural networks for intelligent attack detection somewhere along the direction using DL methods
Chi-Hsuan Huang et al.	SDN	IDS, adversarial attacks	Simulation	JSMA—35%	Studies on malicious examples for deep learning detection systems based on SDN	DDoS, U2L, and R2L are examples of adversarial attacks on SDN-based deep learning IDS systems
Wooyeon Jo et al.	CNN	IDS	NSL-KDD	90%	Three methods for CNN preprocessing	Adding one field to one pixel through preprocessing is highly beneficial for convolution learning, according to similar research that transformed data packets into binary

(continued)

Table 1 (continued)

Author	Algorithm/protocol	IoT security problem	Implementation	Accuracy	Proposed method	Future work
Manimurugan et al.	Deep belief network	IDS	CICIDS 2017	99.37%	We were using the CICIDS dataset for the detection of threats in the DL model DBN-IDS scheme	The based intrusion detection framework can be expanded to detect certain forms of attacks against IoT devices, as well as a variety of intrusion prevention datasets
Alaeddine Boukhalfa et al.	LSTM	NIDS	KDD Cup 99, NSL KDD, DARPA IDS	99.98%, 99.93%, FPR-0.068%, 0.023%	A new NIDS concept based on the Deep Learning system LSTM will identify attacks and hold a long-term memory of them together to order to prevent certain new attacks while also treating all types of attacks in a special way	We want to use our new proposed deep learning model LSTM to introduce a new smart NIDS in the modern world

3 Methodology

3.1 WBC Smart Infrastructure

In the framework of a smart infrastructure environment, an experimental IoT system is implemented with WBC. The smart infrastructure IoT system smart home testbed is depicted in Fig. 4. Cameras, electrical devices an Arduino, a Wi-Fi ESP8266 component, and a gateway computer hardware are all included in the design. An ambient street light sensor, hazardous proximity sensor, temperature sensor, fire sensor, pressure sensor, humidity sensor, and sound sensor, gas sensor, are used to sense the environmental parameters. A person entering and leaving a smart building is detected using directional microphones. The following are the functions of smart infrastructure.

1. Relay 1: Throughout the day, when there are persons inside the smart building and also the environmental street light will be less than the threshold value, the smart building is locked.
2. Relay 2: Whenever individuals are located within the building and also the temperature exceeds a threshold value, the door is locked.
3. Whenever a fire or gas leak is detected, the beep sounds an alarm.
4. When a sound reaches and no one is within the smart building, the light illuminates to identify burglary.
5. The sensor module (ACS712) detects the present and transforms it to a reference voltage that is relevant to the situation (0 and 5 V). The voltage sensor (ZMPT101B) detects voltages ranging from 0 to 1000 V air conditioning. The

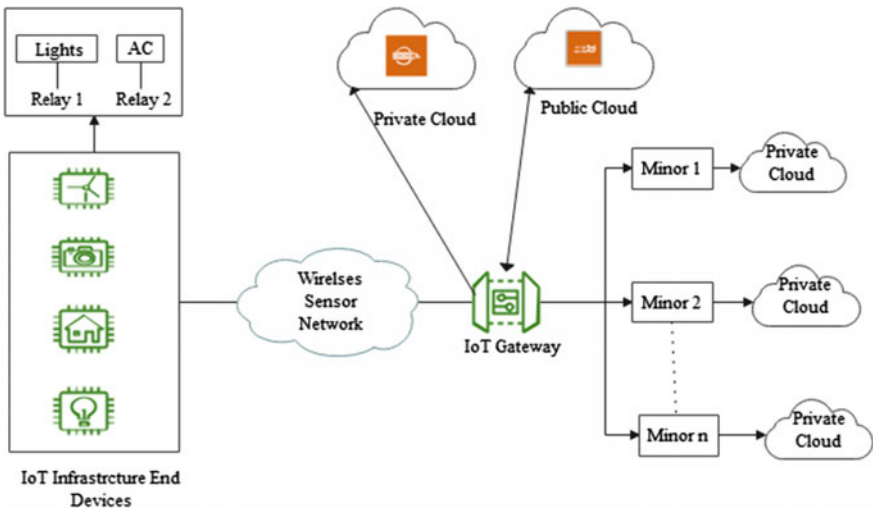

Fig. 4 Smart infrastructure WBC

smart infrastructure energy consumption is determined to use the voltage and current sensor values.
6. The smart building state is processed using UNIX time obtained from the remote system.
7. The TCP/IP protocol is used to transport sensor information from the network edge to the gateway via the ESP8266.
8. The detected key is protected to use the AES method with PBKDF2 and safely saved using WBC at the system level.

3.2 WBC Parameters

The hash function in a bitcoin system is set to 1 MB, and then a block is mined every 10 min [22]. WBC uses a configurable key length with a maximum of 1 megabyte. If another block size is greater than 1 MB, then data is split into many blocks. The data block is determined by the following formula:

$$\text{Size of the Block} = ((\text{Hash Encryption} + \text{Data})\text{Samples} + \text{Size overhead}) \quad (1)$$

When the hash is used encrypted inefficiency is the amount of time it takes to data encryption in bytes. The amount of data examples is represented by tests, while the number of blocks utilized to represent a block is represented by block overhead. The block capacity is restricted to <1 MB in this case. The complexity of determining a target value is determined by the following factors [24].

$$\text{Complexity} = \frac{\text{Complexity\_level\_Target}}{\text{Current\_target}} \quad (2)$$

The time it takes to create a block is determined by using the following formula.

$$\text{Avg Length of Time} = \frac{\left(\text{Complexity} 2^{\text{Bits}}\right)}{\text{Hash Speed}} \quad (3)$$

The difficulty level of mining is represented by bits, while the hash rate is indeed the amount of hashes miners calculate every sec.

3.3 The Infrastructure Device Security is Model as Follows

At step S1, a traditional data cryptography algorithm E receives a security password P as input and produces a signature K public randomly. Using public-key cryptography K-Public and device data D, this encryption method produces a data stored encryption data D at phase S2. The smart contract stores its public key K-public using varied

access permissions also for miners. Figure 4, as inputs, where n is the window frame size. Otherwise, the m miners start solving a POW puzzle by multiplying the nonce number by one for each repetition, unless the BlockID of the block header already exists inside any block within the ledger. If the riddle is resolved and also the miners validate it, e WBC algorithm 1 is successful.

Append-L is seen in algorithm1 (WBC), that also takes a current block's, previous block hash, BlockID, timestamp, edge ID Parts, nonce, MinerID, smart contract hash, and encrypt data, as well as block hash, Edge ID, timestamp, bits, nonce, previous block hash, Block ID, Miner ID, smart contract hash, and encrypt the data. Encrypted as input, data from the previous $(n-1)$ blocks, where n is the window's size. If the BlockID of the current layer already exists in any chain in the ledger, the process crashes; else, the m miners start solving a POW problem by adding the nonce amount using every additional iteration. Because an attacker does not specify the window size, computing hashes for all potential window block sizes is required to carry out the attack effectively. As a result, the computation time rises to

$$t_c = \mathrm{O}(n_t \times tn_t) = \mathrm{O}(n^2 2^b) \tag{4}$$

Let information be the combination of blacklist such as timestamp, Merkle tree, and encrypted data. The present block to be processed is represented by B_{curr}, with data is comprised of the concatenated of block attributes such as BlockID. Every nonce of the current block is represented by current, just as it is by N is then used to calculate the block hash of the current block H below Eq. (5).

$$H_{curr} = h\left(\sum_{i=1}^{n-1} B_{l-i} + [B_{curr} + N_{curr}]\right) \tag{5}$$

Algorithm 1: Append-L

Data: $h(a)$— Function that computes hash of input a

$N_{curr} = 1$
while *True* **do**
  $H_{curr} = h\left(\sum_{i=1}^{n-1} B_{l-i} + [B_{curr} + N_{curr}]\right)$
  if H_{curr} *achieves target* **then**
    break
  else
    $N_{curr} = N_{curr} + 1$

3.4 Deep Learning with IoT Security Model

The algorithm is used to simulate humpback whale hunting behavior, four devices using a spiral to simulate the attacking process and for correct search agent. Encircling the prey, using the bubble-net attacking (BNA) approach, and seeking the prey are the three phases of the mathematical model. LSTM has many benefits; it has many disadvantages when dealing with large datasets. This necessitates the use of a large number of memory cells, which increases computational complexity. This type of problem can also lead to computational complexity. To reduce this issue, you'll need a computationally model for predicting the various types of visible and invisible IoT attacks. To meet the above requirements, a more developed, quick, reliable, and accurate model is needed. The current study aims to establish new hybrid algorithms for LSTM networks [13–16]. The whale optimization algorithm is a new schema optimization technique that mimics humpback whale intelligence bubble-net fishing activity. It is an easy, powerful, and predator probabilistic optimization algorithm that can avoid local optima and find the global optimal answer. The whales begin by encircling their prey with spiral-shaped bubbles that extend down to 12 feet below the surface. And they swim back up to catch and capture their prey, which can be mathematically interpreted by updating old approaches rather than picking the best by random selection of new solutions. It differs from other optimization algorithms in that it only requires the adjustment of two parameters. This number of criteria allows for a seamless transition between the extraction and exploration processes [17] Fig. 5.

$$X(t+1) = \text{Dl.ebl.}\cos(2\pi l) + X*(t) \tag{6}$$

D in the latest millennium is the gap between both the new role and the modified position b is constant which varies from the 0 to 1. $X(t+1)$ is the most advantageous position in the present circumstances. The weights of the LSTM network are optimized using metaheuristic algorithms, and the validity of the model is referred to as the fitness function. Input bias and weights are determined for each iteration. These amounts are then fed into the LSTM network that calculates fitness. If the fitness value is equivalent to the threshold, the iteration will either come to a halt or continue. Whale optimization has a slower convergence speed than most other metaheuristic algorithms, but it takes less time to optimize and increases time consumption [18–20].

4 Results and Discussions

4.1 WBC Experiment Results

The experiment was carried out on an Intel Core i7 personal computer with a Windows 10(8th generation) operating system, 16 GB of RAM, and an Arduino Uno with

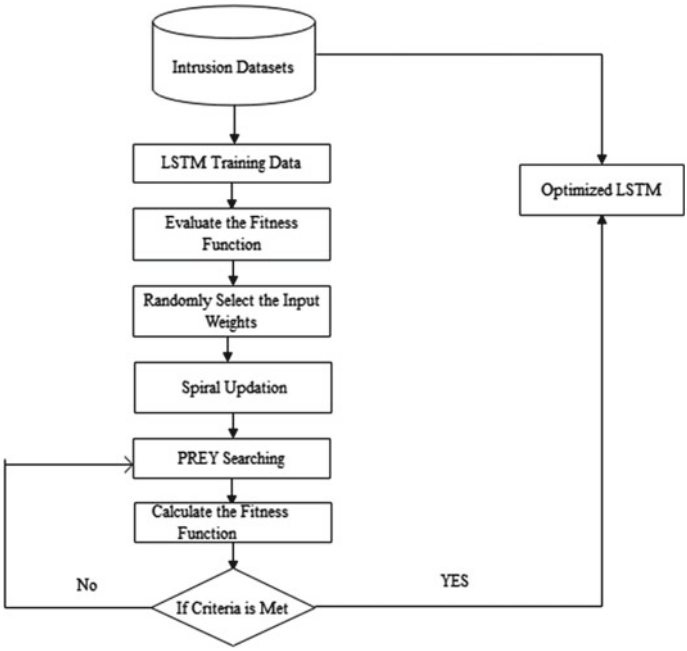

Fig. 5 Proposed WILS training and testing model

digital and analog pins to which cameras are connected serves as end devices. Sensors are connected to the 3.3 V ports of Arduino, whereas switches are connected to 5 V. The sensor connections are linked to the Arduino's input pin first, Vcc (3.3 V) second, then GND third. The Arduino Uno's Tx and Rx are wired to the ESP8266's Serial port. The AT instructions on the ESP8266 are used to connect with both the TCP/IP interface. The ESP8266 is attached to a shared entry point and connects with the computer through it. It requires about 14 s as an Arduino Board to charge up, confirm the ESP8266's availability, and establish a fresh TCP/IP communication with the router [21–23]. The above experiment is also conducted on Arduino Due [24] as edge device and Raspberry Pi as the blockchain miner.

Documents of various sizes are generated as a result of the impact of the threshold value. A 1 megabytes file represents a single block, while a ten-gigabyte file represents ten blocks. The time that it takes to generate a data file hash cost is computed. The rise in window size does have a sufficient level on the hash time complexity of a WBC, as shown in Fig. 6. As regards the window size, overall hash. Processing time is a continuously rising function. $F(x) = b + mx$ Where $f(x)$ is the time it takes to compute a hash, m is the window size, and $b = x$, the slope is the constant. Figure 6 shows this. The slopes are hash computation time (HCT) is $b = 0{:}01{,}641$ and $m = 0{:}0052$. As a result, the moment it takes to compute a hash can be stated as the Fig. 6 HCT = windowsize + 0.01641 × 0.0052. Therefore, WBC increases blockchain security with a negligible increase in computational complexity. WBC

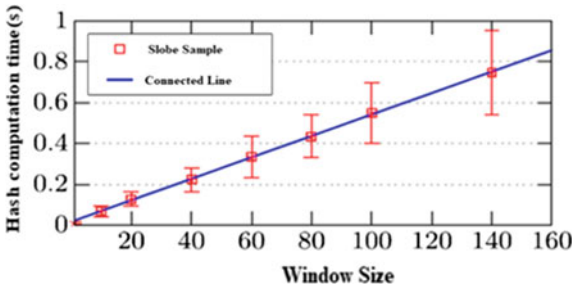

Fig. 6 Analysis of hash computation time versus WBC size

needs a mechanism to store the sequence of window sizes used for each block generation. Our analysis of WBC is carried out with fixed window size. The window size can be chosen based on: (1) our application; (2) requirement of time taken to mine a block; and (3) the required security level. WBC takes less amount of time to mine a block with small window size.

As a result, WBC improves blockchain security while increasing powerful cryptographic only a little. The size of a window might be constant or adjustable. The blockchain is now more secure and harder to compromise with changing window sizes. A dynamic window size calculates the hash of each incoming block in the blockchain using varying window sizes [25]. A changing WBC also requires a technique to keep track of the series of evaluation metrics utilized by each block generation. The WBC analysis has been done with set window size. The length of a window can be determined by our software, the time taken to mine a block, and the level of security needed. With a tiny sliding window, WBC takes less time to mining a block.

Changes in the network traffic, this number may change. The confirmation times for both the WBC with PC, WBC using bitcoin blockchain on Raspberry Pi, PC with a single miner are 92.27 s, 1098.80 s, and 9.16 s, respectively, with a 20-bits difficulty Fig. 7. For Arduino ATmega-based boards, the highest data capture frequency is around 10,000 times a second (nano, mini, and mega, UNO). Since a result, for fast real-time IoT applications, size of window 10–20 MB is preferred, because it has a hash computational cost of 0.066 and 0.126 s, accordingly.

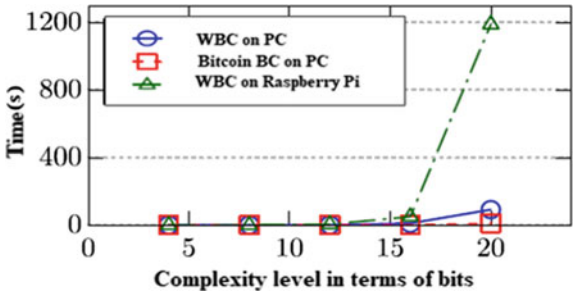

Fig. 7 Time taken for validation

4.2 WLSTM Simulation Results

The experiment was carried out on an Intel Core i7 personal computer with a Windows 10(8th generation) operating system, 16GB of RAM, and a graphics card. The OMNET++IoT API was used to construct a real-time scenario, and the threat model was generated using Python programming. TensorFlow version 1.3.5 was used to run the proposed deep learning system. Pandas is a data analysis and features extraction tool used the dataset NSLKDD-41, CIDDS-001, UNSWNB15 attributes with a single sticker. The proposed Fig. 9 accuracy is found to be 99% accurate and stable whenever the iterations are optimized to 50. Moreover, between training and testing verification, the root means square error is smaller than 0.0001. The suggested model's training and testing accuracy vary from 98.5% to 99%, with RAMSE errors ranging from 0.001 to 0.004. The suggested model's efficiency shows Fig. 8 comparable characteristics when tested using real-time metrics, demonstrating it can predict various.

UNSW-NB15-49 elements with one classification results, 56,000 examples of regular traffic, and 119,341 cases of attacked traffics were used for learning while the

Fig. 8 Proposed model prediction attack analysis

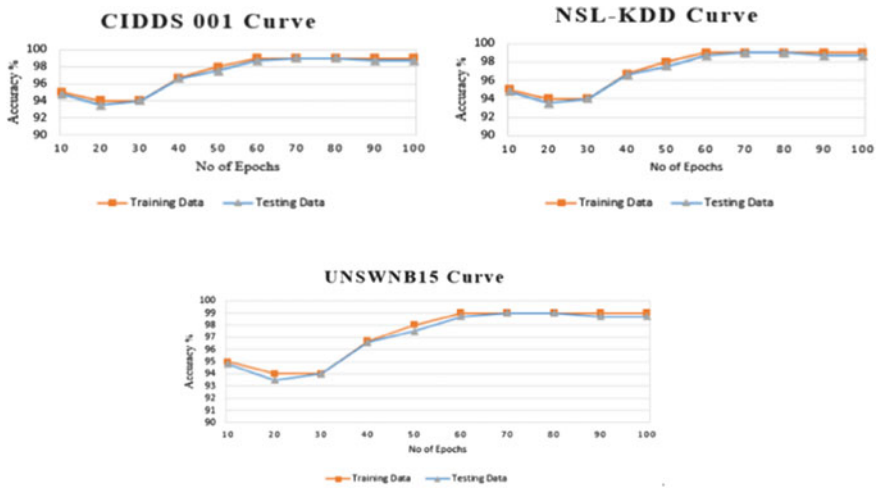

Fig. 9 Proposed model prediction and dataset analysis

testing process consisted of 37,000 instances of traffic flow and 45,332 occurrences of attack traffic, NSLKDD-41 elements with one label attributes Fig. 9.To back up previous studies, the setup was used to record and collect roughly three months of regular and assault data packets. Python API has been established to insert the many sorts of IoT network attacks generated to collect the malicious data from the configuration. In a nutshell, 45 days of regular data and 45 days of malicious activity were logged as separate occurrences and saved in log files. It is subsequently utilized to categories the data.

5 Conclusion

IoT device resources like computing requirements, sources of energy, and memory are all limited. Traditional security techniques are unsuitable for IoT. Developed a WBC that reduces memory cost and limits computation complexity to satisfy the requirements of a resource-constrained IoT network only a fraction of the blockchain is kept in the IoT device, as indicated by the window size, while the complete optimized blockchain is kept in the hybrid network. That uses a difficulty scale of 1–5 while omitting the Merkle tree to minimize processing time. The properties of n blocks in the windows are used to build a block size of the hash which improves security. A fraudulent miner will need the previous $(n - 1)$ blocks as well as the window size data to construct a block. Deep learning model was presented using WILS system. A large number of the dataset were gathered using a real-time scenario using OMNET++IoT plugins, and a Python API is created to insert various malicious activity through networks. The proposed model's output has been tested and related

to other deep learning utilizing benchmark datasets such as CIDDC-001, UNSWN15, and KDD datasets, as well as actual datasets. In all of the tests, the proposed whale optimized in LSTM outperformed other algorithms by 99% in terms of effectiveness, precision, and recall when distinguishing malicious activity in an IoT system and detecting unknown attacks. Based on performance results. In future work, the influence of a varying WBC can be studied, to fit the IoT environment, new consensus methods can be devised. Moreover, the energy consumption of the blockchain can be investigated in order to acquire a deeper understanding of the energy supply required by an IoT device. Whale algorithm is a new form of swarm intelligence algorithm (SIA); the next step is to integrate LSTM in SIA refine the algorithm so that it can be used in more resolve issues in IoT devices with more constraints.

References

1. Zhihan Lv, Bin Hu: Infrastructure monitoring and operation for smart cities based on IoT system. IEEE Trans. Indust. Inf. **16**(3). https://doi.org/10.1109/TII.2019.2913535 (2020)
2. Mukherjee, A., Chakraborty, N.: Whale optimization algorithm: an implementation to design low-pass FIR filter. IEEE [IPACT2017]. https://doi.org/10.1109/IPACT.2017.8244929
3. Dorri, A., Kanhere, S.S.: LSB: a lightweight scalable blockchain for IoT security and anonymity. J. Parallel Distrib. Comp. https://doi.org/10.1016/j.jpdc.2019.08.005.
4. Khraisat, A., Alazab, A.: A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity. https://doi.org/10.1186/s42400-021-00077-7 (2021)
5. Vinayakumar, R., Alazab, M.: A visualized botnet detection system based deep learning for the internet of things networks of smart cities. IEEE Trans. Indust. Appl. https://doi.org/10.1109/TIA.2020.2971952 (2020)
6. Monikaroopak, Tian, G.Y.: An intrusion detection system against DDos attacks in IoT networks. IEEE. https://doi.org/10.1109/CCWC47524.2020.9031206 (2020)
7. Su, T., Sun, H.: Bat: deep learning methods on network intrusion detection using NSL-KDD dataset. IEEE. https://doi.org/10.1109/ACCESS.2020.2972627 (2020)
8. Tanzir Mehedi, S.K., Shamim, A.A.M.: Blockchain—Based security management of IoT infrastructure with ethereum transactions. Springer. https://doi.org/10.1007/s42044-019-00044-z (2019)
9. Liang, W., Huang, W.: Deep reinforcement learning for resource protection and real-time detection in IoT environment. IEEE Internet Things J. Doi: https://doi.org/10.1109/JIOT.2020.2974281 (2019)
10. Ly, V., Nguyen, Q.U.: Deep transfer learning for IoT attack detection. doi: https://doi.org/10.1109/ACCESS.2020.3000476 June 18, 2020
11. Manimurugan, S., Almutairi, S.: Effective attack detection in internet of medical things smart environment using a deep belief neural network. IEEE. doi: https://doi.org/10.1109/ACCESS.2020.2986013 (2020)
12. Ning, G.-Y., Cao, D.-Q.: Improved whale optimization algorithm for solving constrained optimization problems. Hindawi vol. https://doi.org/10.1155/2021/8832251 (2021)
13. Patidar, S., Singh Bains, I.: Web security in IoT networks using deep learning model. IEEE Xplore. doi: https://doi.org/10.1109/ICSSIT48917.2020.9214114 (2020)
14. Soleimanian Gharehchopogh, F., Gholizadeh, H.: A comprehensive survey: whale optimization algorithm and its applications. https://doi.org/10.1016/j.swevo.2019.03.004 (2019)
15. Beltagy, I., Peters, M.E.: Longformer: the long-document transformer. https://github.com/allenai/longformer (2020)

16. Jiang, F., Fu, Y.: Deep learning based multi-channel intelligent attack detection for data security. IEEE Trans. Sustain. Comp. **5**(2). doi: https://doi.org/10.1109/TSUSC.2018.2793284 (2020)
17. Boukhalfa, A., Abdellaoui, A.: LSTM Deep Learning Method for Network Intrusion Detection System, vol. 10, no. 3 (2020)
18. Otoum, S., Kantarci, B.: On the feasibility of deep learning in sensor network intrusion detection. IEEE Netw. Lett. **1**(2). doi: https://doi.org/10.1109/LNET.2019.2901792 (2019)
19. Reyna, A., Martín, C.: On blockchain and its integration with IoT. Challenges Opportunities. https://doi.org/10.1016/j.future.2018.05.046 (2018)
20. Waheed, N., He, X.: Security and privacy in IoT using machine learning and blockchain: threats and countermeasures. ACM Comput. Surv. **53**(3), (2020)
21. Riyanto, R.A.: Discretizing whale optimization algorithm to optimize a long short-term memory. IEEE. doi: https://doi.org/10.1109/ICOIACT50329.2020.9331972 (2020)
22. Sun, X., Jin Gao, X.: The research on application of sliding window LS_SVM in the batch process. doi: https://doi.org/10.1109/ACC.2013.6579852 (2013)
23. Rana, N., Latiff, M.S.: Whale optimization algorithm: a systematic review of contemporary applications, modifications and developments. https://doi.org/10.1007/s00521-020-04849-z (2020)
24. Fu, J., Qiao, S.: A study on the optimization of Blockchain Hashing algorithm based on PRCA. https://doi.org/10.1155/2020/8876317 (2020)
25. Sultan, M.A.M.: IoT Security Issues via Blockchain: A Review Paper. doi: https://doi.org/10.1145/3320154.3320163