Noname manuscript No.
(will be inserted by the editor)

Deep Neural networks based TensorFlow Model for IoT lightweight cipher attack

Zakaria TOLBA · Makhlouf DERDOUR

Received: date / Accepted: date

Abstract The internet of Things (IoT) technology is present in all aspects of our modern lives, and its standard usage is increasing remarkably. But their inherent limitations in size, storage memory, and power consumption limit its specific functionality in the secure transmission of sensitive information, where the development of lightweight ciphers responds adequately to these limitations. However, the conventional cryptanalysis of these modern ciphers can be impractical or demonstrate apparent limitations to be generalized. Because they frequently require a large amount of considerable time, known plain texts, and big storage memory, they are typically performed without the restriction of key space, or only the reduced round variants are attacked. This work proposes a deep learning (DL) model-based approach for a successful attack that discovers the plain text from cipher text one, it's demonstrated that the proposed DL-based cryptanalysis represents a promising step towards a more efficient and automated test to verify the security of emerging lightweight ciphers. We directly attack the encryption independently of the key or the number of rounds using the TensorFlow platform in google collaboratory notebook environment that runs in the cloud and stores the results on Google Drive, the results are communicated to demonstrate precisely the effective performance of the attack, and numerous experiments were performed to confirm the study.

Zakaria TOLBA
The Laboratory of Mathematics Informatics and Systems (LAMIS)
Larbi tebessi University - Tebessa
Algeria
E-mail: zakaria.tolba@univ-tebessa.dz

Makhlouf Derdour
The Laboratory of Mathematics Informatics and Systems (LAMIS)

Keywords Tensorflow · Deep learning · neural networks · Cryptanalysis · lightweight cipher · attack · Internet of Things

1 Introduction

Nowadays the world marks the contemporary era of IoT, where all data travel from personal device to another along with personal and confidential information. This specific information ordinarily requires private security. Cryptography, this modern art of scientifically investigating the sophisticated techniques for properly securing sensitive information either in communication networks or in proper storage data [1].
The famously used cryptograms like AES [11] and DES [3] critically require an important number of necessary resources for their successful implementation, these cryptograms are unfeasible in the specific IoT devices [17] [6] [9] [18] because of the potential limitations of various performance metrics. Lightweight cryptography occupied a principal role in handling devices that have typically limited memory space to overcome these fundamental limitations.
Lightweight block ciphers operate correctly for IoT on a specific block of sophisticated data for fixed-length specific bits and a symmetric key with a profound transformation. In general cases, these profound transformations in common remain elementary operations with bits, like possible substitution and permutation networks (SPN) or private Feistel networks.
Lightweight ciphers efficiently are mostly symmetric ciphers, made lightweight in practical terms of modest size, small storage, local memory, potential limited energy, and processing time, it's popularly used in smart healthcare sensors, intelligent wireless multimedia surveillance networks (SWMSN), radio-frequency identification tags (RFID), self-driving vehicles (SDV), drones surveillance systems (DSS) modern cars, bio-chip remote farm animals surveillance, cyber-physical systems (CPS), the intelligent transportation system (STS), and smart industry 4.0 monitoring system, etc.
Cryptanalysis is an audit step that leads designers to develop more robust cryptographic algorithms and adequately assess algorithms' overall impressive performance. However, The fundamental problem is that cryptanalysis of these ciphers can be impractical or convincingly demonstrate apparent limitations to be generalized. Because they frequently require a large amount of considerable time, known plain texts, and big storage memory, they are typically performed without the arbitrary restriction of keyspace, or only the reduced round variants are typically attacked.
This practical work proposes a deep learning (DL) model-based approach for a successful attack on KATAN 32 Bit that discovers the plain text from cipher text one, it's representing a promising step towards a more efficient and automated test to verify the security of emerging ciphers. We directly attack the encryption independently of the private key or the used number of rounds utilizing the TensorFlow platform in a google collaboratory notebook environment that runs in the cloud and stores the results on Google Drive.

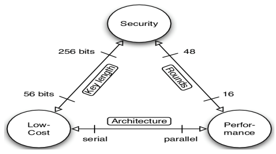

Fig. 1 Trade-off between Security,Performance and Cost of IoT devices

2 IoT Lightweight Ciphers Implementation Techniques

IoT lightweight ciphers are implemented by two following techniques:

• Hardware Implementation:

In this apparent case, the algorithms are efficiently implemented in specific hardware [2], the practical efficiency of the primitive is adequately evaluated by the following metrics:
– Gate Equivalents (GEs) which define the physical memory area ordinarily required to efficiently implement the algorithm primitive. The impressive performance will be excellent if the specific area is lesser.
– The latency traditionally remains the proper time instantly conducted by the hardware circuit to efficiently produce the specific output. It is correctly valued in necessary seconds. The remarkable performance will be more satisfactory if latency is lower.
– Energy consumption efficiently is the economic power consumed by the specific hardware circuit, The economic performance is good for the low power consumption .

• Software Implementation: The lightweight algorithm can also be implemented in specialized software [6], mainly for practical use on micro controllers. In this specific type of successful implementation, the performance metrics which will in common be properly evaluated correctly are the Random Access Memory (RAM) usage, program length, and throughput.
–The throughput efficiency represents the measurable quantity of the authentic message being properly processed per time unit. It is valuated as bps. the optimal performance will be more impressive if the private message is typically processed highly.
– Random Access Memory (RAM) usage related to the reasonable amount of necessary information carefully scripted to the proper storage IoT specialized device.
– Program length represents the determinate quantity of necessary information ordinarily required to validate the economic performance without typically depending on its essential input .

The figure 1 highlights the necessary trade-off between Cost, Performance, and marketable private security. The impressive performance of lightweight cryptography is critically evaluated by the used metrics like latency, energy consumption, throughput, and typically waits for the proper time.

3 RELATED WORK

The practical use of machine learning in modern cryptography is not new but successful works in cryptanalysis have scarcely recently emerged. The most famous use case of machine learning is distinguishing or reasonably the identification of encryption algorithms typically based on ciphertext data only.
These published papers have been efficiently implemented to objectively analyze specific data encrypted by modern block ciphers such as AES, 3DES, Blowfish, and Camellia . Cryptanalysts have also attempted to use efficiently deep learning straightforwardly - performing decryption of ciphertexts without any practical knowledge of the private key. This is equivalent to training a machine learning model to emulate or mimic an encryption algorithm for a static private key. For this specific purpose [2] implemented unsupervised learning with deep neural networks to typical cryptanalysis substitution ciphers such as the Vigenere and Shift ciphers.

After that, Mishra et al. investigated whether neural networks can be intentionally used to correctly predict the chosen plaintext of PRESENT block cipher from any particular round [7]. Xiao et al. reasonably described a black-board security evaluation approach to accurately measure the considerable strength of proprietary ciphers without the necessary knowledge of the encryption algorithms themselves [12].
They precisely quantified the effective strength of a modern cipher by measuring how difficult it was objectively for a neural network to mimic the cipher algorithm. Their published results instantly showed that the marketable security of the modern cipher used for key less entries in modern cars appropriately named Hitag2 was weaker than 3 DES. In [8] Cipher mimicking has also been explored. The academic researchers optimized a deep neural network to decrypt ciphertexts of 64-bit DES.
Perov convincingly demonstrated that using deep-learning techniques could accurately distinguish the ciphertexts of round-reduced ciphers from arbitrary sequences [13].
A deep learning model was used indoors [14] to positively predict the desired outputs of a quantum random number generator.

Gohr properly introduced what was considered functionally the first successful application of machine learning from the comparative perspective of conventional cryptanalysis. A successful machine learning-based differential distinguisher in a side-channel attack was adequately developed and properly used to typically attack the lightweight block cipher Speck32/64 reduced to 11 rounds.

deep learning has also been applied to traditionally perform and objectively evaluate the linear cryptanalysis on DES based on linear expressions [15].
[16] recently proposed machine learning classifiers that can correctly classify differential trails as secure or insecure based on differential private data. Many practical experiments were performed impressively on GFS ciphers. Their pub-

lished papers satisfactorily showed that the well-trained models were able to generalize to modern ciphers that the developed models have not seen in the past. Finally [17] attempted key recovery attacks on block ciphers, Simon and Speck, using deep learning for side-channel attacks. their published work was successful in properly recovering encryption secret keys for full-round Simon32/64 and Speck32/64 only when the key space of the modern cipher was traditionally restricted to text-based keys.

4 TensorFlow and TensorBoard Framework

TensorFlow intellectually represents an ethical Open Source framework that ideally allows us to gently apply machine learning and efficiently perform various complex calculations on decentralized data.TensorFlow was adequately developed to promote open science and creative experimentation with federated learning, a machine learning method by which a shared global model is operated by a considerable number of voluntarily participating clients who maintain their training data locally.

TensorFlow not merely allows independent developers to accurately simulate the federated learning algorithms properly included on their developed models and private data but also to test new algorithms. The fundamental components generously provided by TensorFlow can be intentionally used to implement correctly not intended for training, such as aggregate analyzes .

TensorBoard provides the visualization and tooling needed for machine learning experimentation like tracking and visualizing metrics such as loss and accuracy and visualizing the model graph and viewing histograms of appropriate weights, biases, or other tensors as they change over time. TensorBoard's Graphs dashboard in common remains an effective tool for comprehensively examining any TensorFlow model. It ideally allows the brief view of a conceptual graph of the appropriate model's structure and ensures, it matches the intended design. we can also view an op-level graph to instantly interpret how TensorFlow recognizes our comprehensive program. Comprehensively examining the op-level graph can give insight as to how to voluntarily revise your developed model. For a notable example, we can redesign our developed model if typically training is progressing slower than reasonably expected.

5 Methodology and Results

We have carefully selected to experiment with the fully connected deep neural network architecture in our regression task. After experimenting with various neural network architectures like Deep believe neural networks (DBN), Auto encoders, RNN, LSTM and CNN and MPLs, we found that the fully connected neural network performed consistently for our prediction problem. In notable addition, a fully connected neural network does not require any fundamental assumption to the input, therefore making it flexible to be correctly applied in our problem.

A fully connected neural network consists of a series of fully connected layers, where each neuron is connected to all neurons in the following layer. Figure 3 illustrates the fully connected neural network that was used in our experiments.

5.1 PROBLEM FRAMING

The global goal of the proposed work is to train neural network models to predict the plaintext from the ciphertext one. Using supervised learning, we framed the problem as a regression task because the goal is to predict the Block cipher consisting of non-negative integers. Practical experiments were performed professionally for both KATAN 32 bit cipher. The ultimate goal of a cryptanalyst is to minimize the distinguisher model from a different block cipher data inputs .

5.2 Datasets processing

By properly using the Katan 32 bit ciphers for proof-concept experiments, we could typically generate large datasets within a practical amount of considerable time. KATAN block cipher is highly compact and achieves indeed more minimal size with a visible footprint of 802 GE compared to PRESENT cipher. It's a hardware-oriented block cipher, with a specific kind of fundamental Feistel structure and uses a simple key scheduling mechanism. It properly includes three distinct variants of 32 bit, 48 bit, or 64-bit block sizes with a key length of 80 bits and 254 possible rounds of active operations. The software implementation is ordinarily found to be inefficient. The cipher eagerly consumes too much potential energy and has low throughput.

We efficiently generated datasets of 1 470 000 samples profitably using Python lightweight cryptography tools in the google collaboratory environments for the training step and 245 000 samples for the model validation.

5.3 NEURAL NETWORK ARCHITECTURE

In all of the following experiments, we utilized fully connected neural networks. We initially performed hyper-parameter tuning to positively identify the optimal number of layers, number of neurons per layer, loss function, optimizer, number of epochs, and batch size for the regression task.

Scientifically based on our practical experiments, we carefully selected a neural network with seven hidden layers. The specific number of neurons per distinct layer differs typically depending on the specific layer. For Katan 32

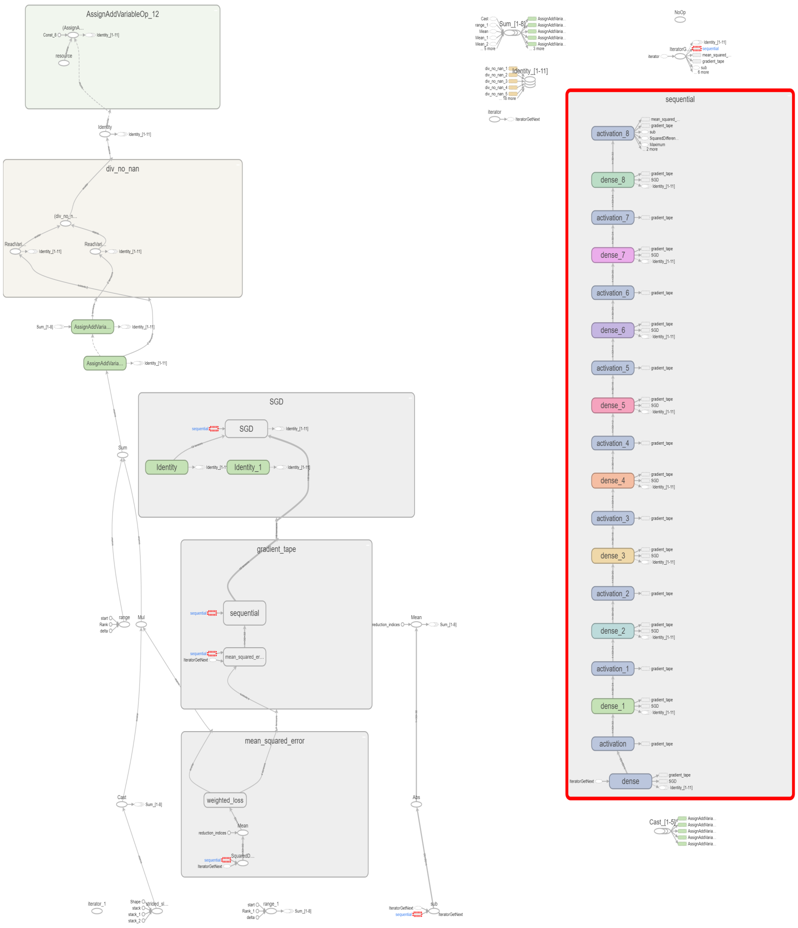

Fig. 2 TensorBoard Graph dependencies for the trained model.

bit ciphers, there are 32 neurons in the input layer (equivalent to the specific number of input features), seven hidden layers with 24,20,16,12,16,20,24 neurons each, and an output layer with 32 neurons to represent the predicted plaintext as depicted in Figure 4. The remaining hyper parameters properly used in our remarkable experiments are summarized below:

- Weights initialization : Glorot Uniform .
- Optimizer: SGD for Stochastic gradient descent Optimizer.
- learning rate = 0.001
- Activation Function: Rectified Exponential linear unit (RELU)
- Epochs: 3000 epochs

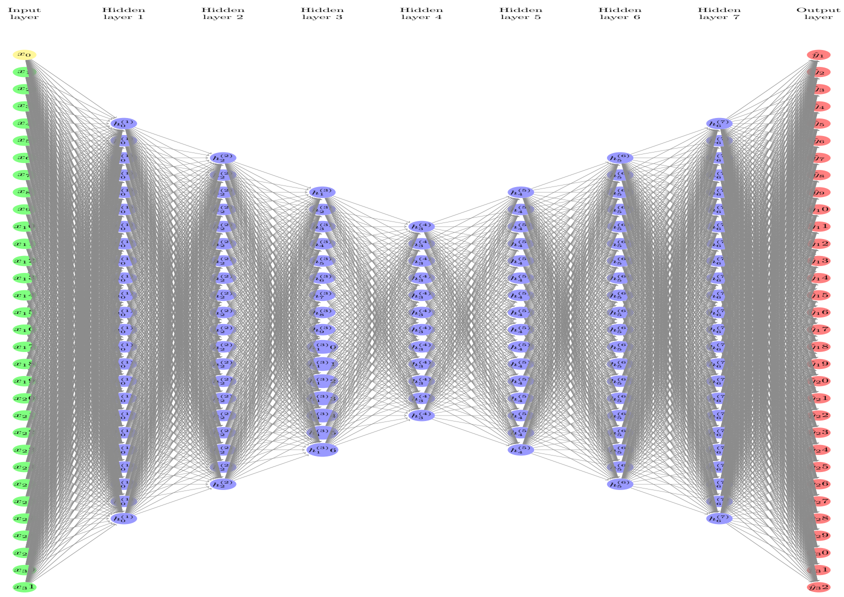

Fig. 3 The fully connected deep neural network Trained model

- Batch Size: 64
- Trainable parameters: 3 748 , None Trainable parameters: 0 .
- Error function: Mean squared error $MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$
- accuracy function : R-squared defined by : $R^2 = 1 - \frac{UnexplainedVariation}{TotalVariation}$

  The R-squared value R^2 is always between 0 and 1 inclusive , An R^2 of 1 indicates that the regression predictions perfectly fit the data.
  Our work is a regression problem the accuracy of the model is defined with these additional metrics:
  – Cosine proximity ,Root Mean squared error , Absolute squared error .
  Figure 2 illustrates the Graph dependencies for the trained model.

5.4 EXPERIMENTAL RESULTS

We have operated the Keras checkpoint named Call Backs to typically save the most satisfactory results after every iteration and to saving the weight and bias of our trained model, After 3000 epoch iterations in 11 hours and 43 minutes, the most impressive results are properly obtained in the 2759 epoch.
Error function: Mean squared error = 0.0087.
accuracy function = 0.89.
The high R-squared value typically ranging from 0.89 also indicates there is

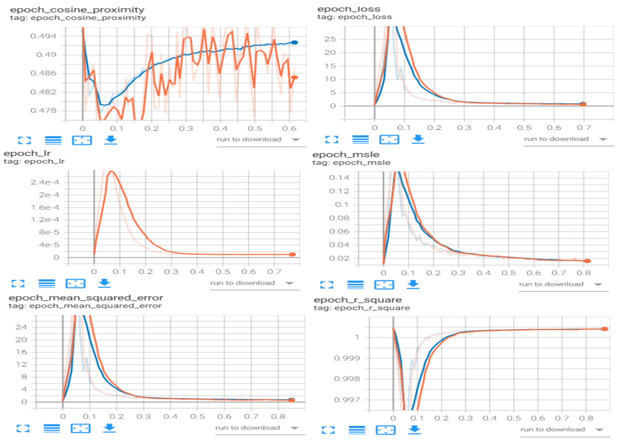

Fig. 4 The experimental results for training and validation respectively of : Cosine proximity , Loss function , Learning rate , Absolute squared error , R-squared , Mean squared error

an effective relationship between predicted plaintext and the ciphertext. The results are illustrated in the following figure 4.

6 CONCLUSION

In this illustrated paper, Our work is properly established as a regression task where we tentatively proposed a deep learning approach to KATAN 32 bit lightweight block cipher security analysis. Exclusively, we typically train fully connected deep neural network models to accurately predict the plaintext from the chosen-ciphertext one. the fully connected deep neural networks are improved using TensorFlow Framework in a Google Cloud environment. We properly investigate the economic feasibility of the proposed attack using cloud tools. in the future works, we plan to extend this work for more larger Block cipher size like KATAN 48 bit and 64 bit , and for more promising step towards a more efficient and automated test to verify the security of emerging lightweight ciphers such as: RECTANGLE, Humming Bird, SIMON, GRAIN, WG-8, ESPRESSO, and TRIVIUM.

References

1. R. B.-R. C. T. LJ and U., "The Electronic Communications Privacy : The Challenge of Applying Ambiguous Statutory Language to Intricate Telecommunication Technologies," Hein On line Act of 1986 .
2. A. Gomez, S. Huang, I. Zhang, B. Li, M. Osama, and L. Kaiser, "Unsupervised Cipher Cracking Using Discrete GANs," in International Conference on Learning Representations, 2018.
3. W. Wu and L. Zhang, "LBlock: A Lightweight Block Cipher," in Applied Cryptography and Network Security. Springer Berlin Heidelberg, 2011, pp. 327–344.

4. K. V. Pradeepthi, V. Tiwari, and A. Saxena, “Machine Learning Approach for Analysing Encrypted Data,” in 2018 Tenth International Conference on Advanced Computing (ICoAC). IEEE, dec 2018.
5. W. Zhang, Y. Zhao, and S. Fan, “Cryptosystem Identification Scheme Based on ASCII Code Statistics,” Security and Communication Networks, vol. 2020, pp. 1–10, dec 2020.
6. F. Yu, X. Gong, H. Li, and S. Wang, “Differential cryptanalysis of image cipher using block-based scrambling and image filtering,” Information Sciences, vol. 554, pp. 145–156, apr 2021.
7. G. Mishra, S. V. S. S. N. V. G. K. Murthy, and S. K. Pal, “Neural Network Based Analysis of Lightweight Block Cipher PRESENT,” in Harmony Search and Nature Inspired Optimization Algorithms. Springer Singapore, aug 2018, pp. 969–978.
8. A. Mundra, S. Mundra, J. S. Srivastava, and P. Gupta, “Optimized deep neural network for cryptanalysis of DES,” Journal of Intelligent Fuzzy Systems, vol. 38, pp. 5921–5931, 2020.
9. G. Bansod, N. Raval, and N. Pisharoty, “Implementation of a New Lightweight Encryption Design for Embedded Security,” IEEE Transactions on Information Forensics and Security, vol. 10, no. 1, pp. 142–151, jan 2015.
10. A. Jain and G. Mishra, “Analysis of Lightweight Block Cipher FeW on the Basis of Neural Network,” in Harmony Search and Nature Inspired Optimization Algorithms. Springer Singapore, aug 2018, pp. 1041–1047.
11. A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, “PRESENT: An Ultra-Lightweight Block Cipher,” in Cryptographic Hardware and Embedded Systems - CHES 2007. Springer Berlin Heidelberg, pp. 450–466.
12. Y. Xiao, Q. Hao, and D. D. Yao, “Neural Cryptanalysis: Metrics, Methodology, and Applications in CPS Ciphers,” in 2019 IEEE Conference on Dependable and Secure Computing (DSC). IEEE, nov 2019.
13. A. Perov, “Using Machine Learning Technologies for Carrying out Statistical Analysis of Block Ciphers,” in 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON). IEEE, oct 2019.
14. N. D. Truong, J. Y. Haw, S. M. Assad, P. K. Lam, and O. Kavehei, “Machine Learning Cryptanalysis of a Quantum Random Number Generator,” IEEE Transactions on Information Forensics and Security, vol. 14, no. 2, pp. 403–414, feb 2019.
15. B. Hou, Y. Li, H. Zhao, and B. Wu, “Linear Attack on Round-Reduced DES Using Deep Learning,” in Computer Security – ESORICS 2020. Springer International Publishing, 2020, pp. 131–145.
16. T. R. Lee, J. S. Teh, J. L. S. Yan, N. Jamil, and W.-Z. Yeoh, “A Machine Learning Approach to Predicting Block Cipher Security,” in Cryptology and Information Security Conference. Universiti Putra Malaysia, 2020.
17. J. So, “Deep Learning-Based Cryptanalysis of Lightweight Block Ciphers,” Security and Communication Networks, vol. 2020, pp. 1–11, jul 2020.
18. E. Biham and A. Shamir, “Differential cryptanalysis of DES-like cryptosystems,” Journal of Cryptology, vol. 4, no. 1, pp. 3–72, jan 1991