future internet

MDPI

Article

ThermalAttackNet: Are CNNs Making It Easy to Perform Temperature Side-Channel Attack in Mobile Edge Devices?

Somdip Dey [1,2,*], Amit Kumar Singh [1] and Klaus McDonald-Maier [1]

[1] School of Computer Science and Electronic Engineering (CSEE), University of Essex, Colchester CO4 3SQ, UK; a.k.singh@essex.ac.uk (A.K.S.); kdm@essex.ac.uk (K.M.-M.)
[2] Nosh Technologies, Colchester CO4 3SL, UK
* Correspondence: somdip.dey@essex.ac.uk or dey@nosh.tech

Abstract: Side-channel attacks remain a challenge to information flow control and security in mobile edge devices till this date. One such important security flaw could be exploited through temperature side-channel attacks, where heat dissipation and propagation from the processing cores are observed over time in order to deduce security flaws. In this paper, we study how computer vision-based convolutional neural networks (CNNs) could be used to exploit temperature (thermal) side-channel attack on different Linux governors in mobile edge device utilizing multi-processor system-on-chip (MPSoC). We also designed a power- and memory-efficient CNN model that is capable of performing thermal side-channel attack on the MPSoC and can be used by industry practitioners and academics as a benchmark to design methodologies to secure against such an attack in MPSoC.

Keywords: multiprocessor system-on-chip (MPSoC); thermal behavior; temperature side-channel attack; security; machine learning; convolutional neural network (CNN); deep learning; energy efficiency; memory efficiency

Citation: Dey, S.; Singh, A.K.; McDonald-Maier, K. ThermalAttackNet: Are CNNs Making It Easy to Perform Temperature Side-Channel Attack in Mobile Edge Devices? *Future Internet* **2021**, *13*, 146. https://doi.org/10.3390/fi13060146

Academic Editor: Georgios Kambourakis

Received: 22 March 2021
Accepted: 25 May 2021
Published: 31 May 2021

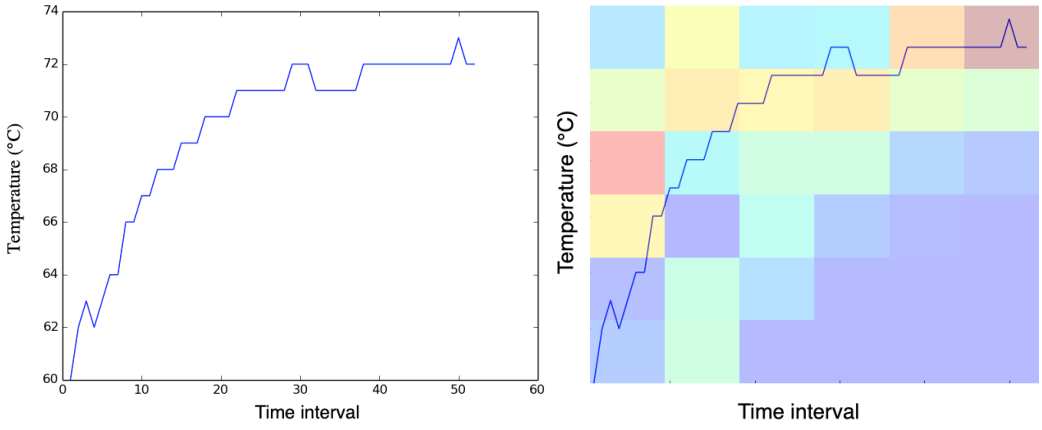
1. Introduction

Recently, mobile devices have become an integral part of daily life. These mobile devices are utilized to run different types of applications, including video calling, web browsing, gaming, navigation; hence, energy-efficient processing on these battery-empowered mobile devices is of utmost importance [1,2]. Mobile cloud computing, where most of the computations happen in the cloud (also known as *Cloud Offloading*) [3], is considered to be a potential solution for energy-efficient processing. However, application processing that needs privacy and security, such as a banking app or a secure data storage app, is often processed on the mobile device instead of cloud offloading. Moreover, as mobile edge computing becomes more and more ubiquitous, security issues in these mobile devices become more paramount. Such mobile devices have to face hostile security threats [4–7], such as physical, logical/software-based, and side-channel/lateral attacks. Amongst these, side-channel attack [4–6] is a popular security threat due to ease of access to the physical hardware where attacks are performed by observing the properties and behavior of the system, such as power consumption, thermal dissipation, electromagnetic emission, etc.

Comparatively, a lot less documented studies are performed in temperature (thermal)-based side channel attacks [4–6] in mobile edge devices. Most of these mobile devices come equipped with heterogeneous multi-processors systems-on-chip (MPSoC), which consists of multiple heterogeneous processors on a single chip, capable of processing different types of applications to cater for performance and energy-efficiency of the executing applications. Due to an increase in the usage of MPSoCs [2,8–13] in mobile edge devices and a rise in studies on thermal side channel attacks [6,14,15], it is crucial that side channel attacks in such a platform should be addressed with utmost importance [15].

To explore feasibility of thermal side-channel attack in a real commercial mobile device, we designed a new type of attack which involved computer vision-based Convolutional

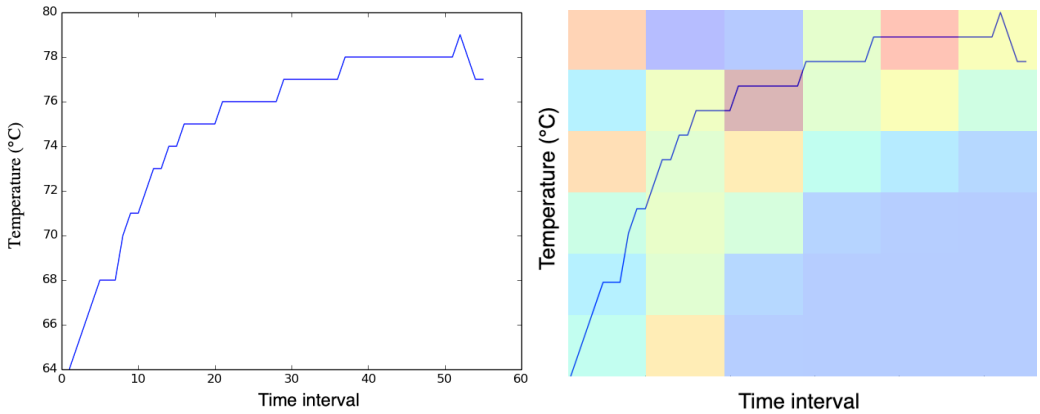
Neural Network (CNN). Among all the fields of Neural Network-based machine learning and pattern recognition, computer vision-based Neural Networks, especially CNNs and Deep Learning (DL) [16,17], are well studied and mature comparatively. Recently, CNN models have achieved high prediction accuracy in applicative fields to solve several real-life challenges, such as traffic categorization [18,19], human rights violation [20], weather forecasting [21], etc. Given the high success rate in understanding patterns, we utilized a CNN model-based attack. To perform the attack, we chose 4 of the 25 most common passwords of 2017 and 2018 [22,23] as surveyed by the Internet security firm SplashData. The 4 common passwords used by the user, which are chosen for our attack, are *123456*, *passw0rd*, *111111*, and *football*. We then executed AES-256 [24] encryption on a text file using the aforementioned passwords on Odroid XU4 development board [25] running on ondemand Linux governor [26] and recorded the thermal behavior of the CPUs. We trained ResNet model [27], a pre-trained CNN model trained on ImageNet using *transfer learning*, with the graphical representation of the thermal behavior (as shown in Figures 1a and 2a). ResNet was able to achieve a training prediction accuracy of 46.88% and a testing prediction accuracy of 31.99%, which means that ResNet is able to predict the correct password, one out of every four attempts on an average. Figures 1b and 2b show the region of interest on the graphical representation of the thermal behavior which is used by the CNN model to predict the password. In order to determine whether ResNet is classifying the thermal data based on the features of the thermal peaks, we utilized Gradient-weighted Class Activation Mapping (Grad-CAM) [28] to visualize in which areas of the graphical data the CNN was focusing on to predict the password being used for encryption process. In Figures 1b and 2b, the area highlighted (heat map) with shades of yellow/red is the active region where the CNN is looking to determine the password used. In the heat map, the regions range from blue to red, where blue means least active region, and red means the most active one. The observations from the aforementioned figures prove that visual-based CNNs could be successfully utilized to perform thermal side-channel attack, and, to the best of our knowledge, this is the first documented study to do so. In summary, this paper makes the following contributions.



(**a**) (**b**)

Figure 1. Focus area of ResNet network on a representative graph of password: 111111.

1. Design and explore thermal side-channel attack using computer vision-based CNN models.
2. Evaluate popular CNN models and their accuracy in predicting password for different Linux governors.
3. Design and implementation of a power- and memory-efficient CNN model, ThermalAttackNet, to perform thermal side-channel attack on a real consumer mobile device.



(**a**) (**b**)

Figure 2. Focus area of ResNet50 network on a representative graph of password: passw0rd.

The main motive to design and implement a computer vision-based CNN model to perform thermal side-channel attack is to provide a benchmark that could be used by industry practitioners and researchers to improve security against such an attack in mobile devices utilizing MPSoCs.

2. Preliminaries

2.1. Convolutional Neural Networks and Deep Learning

A Deep Learning (DL) model [29] consists of an input layer, several intermediate (hidden) layers, which are stacked on top of each other, and an output layer. In the input layer, which is the first layer of the model, the raw values of data features are fed into it. In each of the hidden layers, a mathematical operation called convolution is applied to extract specific features, which is then utilized to predict the label of the raw data in the last (output) layer of the DL network. Most of the time, if a model utilize an input layer, a hidden layer and an output layer then the model is denoted as Convolutional Neural Network (CNN) model or simply, CovNet. If such a model uses a lot of stacked hidden layers, only then it is denoted as a DL model or Deep Neural Networks (DNN).

2.2. Pre-Trained Networks and Transfer Learning

A conventional approach to enable training of DNN/CNN on relative small datasets is to use a model pre-trained on a very large dataset, and then use the CNN as an initialization for the applicative task of interest. Such a method of training is called "*transfer learning*" [30], and we have followed the same principle. The chosen CNN models mentioned in Section 3.3 are pre-trained on ImageNet [29]. For the proposed attack, we have utilized the following popular pre-trained CNN models: VGG (VGG19) [31], ResNet (ResNet152v2) [27], MobileNet (MobileNetv2) [32], and NASNet (NASNetMobile) [33].

3. Thermal Side-Channel Attack Using CNN

3.1. Hardware & Software Setup for Experiments

We also chose Odroid XU4 [25] board to execute the attack in order to verify the affect of thermal side-channel exploitation. Odroid XU4 employs the Samsung Exynos 5422 [34] MPSoC, which is popularly used in Samsung mobile devices, especially Samsung Galaxy S5. The Odroid XU4 is a representational development board of Galaxy S5 smartphone. Exynos 5422 MPSoC contains clusters of big (4 Cortex A-15) and LITTLE cores (4 Cortex A-7). This MPSoC provides DVFS feature per cluster, where the big core cluster has 19 frequency scaling levels, ranging from 200 MHz to 2000 MHz with each step of 100 MHz and the LITTLE cluster has 13 frequency scaling levels, ranging from 200 MHz to 1400 MHz, with each step of 100 MHz.

The Odroid XU4 was running on UbuntuMate version 14.04 (Linux Odroid Kernel: 3.10.105). During the time of performing the attack, the average ambient temperature of the room was 21 °C. When we executed the attack, we changed the governor [26] between conservative, ondemand, performance, interactive, and powersaver to study which Linux governor is more vulnerable to such attack.

Brief description of the different types of governors are as follows:

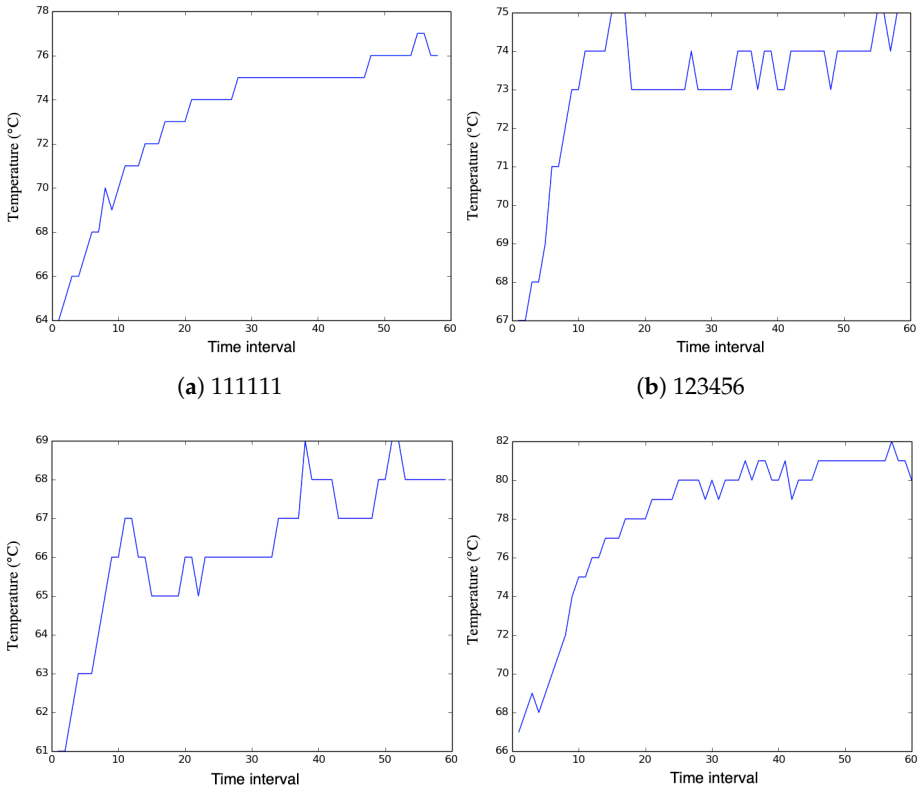
- **ondemand**: Sets the operating frequency of the CPU depending on the CPU utilization. In this, the operating frequency is set to maximum whenever there is any load on the CPU.
- **conservative**: Is a fork of ondemand governor and sets the operating frequency of the CPU depending on the CPU utilization. It differs from ondemand by increasing or decreasing the operating frequency of the CPU gradually based on the CPU utilization.
- **performance**: Sets the operating frequency of the CPU to the highest frequency within the borders of user specified minimum frequency and maximum frequency.
- **powersaver**: Compared to performance, this governor sets the operating frequency of the CPU to the lowest frequency within the borders of user specified minimum frequency and maximum frequency.
- **interactive**: Dynamically scales CPU operating frequency in response to the CPU utilization. Interactive is significantly more responsive than ondemand because it scales the operation frequency over the course of time to max frequency based on the CPU utilization.

3.2. Dataset and CNN Model

To generate a dataset of thermal behavior, we choose 4 most common passwords (*123456*, *passw0rd*, *111111*, and *football*) and used AES-256 encryption algorithm to encrypt a text file using the aforementioned passwords. For each aforementioned password, the encryption was performed on the same text file for 500 times. The reason to choose AES-256 is because of its popularity. The encryption operations were performed on CPU 7, which is one of the big CPUs (A-15) of the Exynos 5422 MPSoC, while one of the LITTLE cores (CPU 3) snoops the operating temperature data of the big CPU. After the temperature data for each password were collected, we transformed the data points into a graphical representation in order to be fed to a pre-trained CNN for training and prediction purposes. Figure 3 shows the graphical representation of the thermal behavior of CPU 7 for 111111 (Figure 3a), 123456 (Figure 3b), football (Figure 3c), and passw0rd (Figure 3d), respectively, during the encryption process.

3.3. Training CNN to Predict Password

We choose a pre-trained CNN model, which is trained on 1000 classes of ImageNet (CNN model is pre-trained with 1000 different labels (classes), such as eskimo dog, madagascar cat, cougar, and lifeboat of ImageNet database.) and removed the classifier module and modified it to be able to predict our chosen classes of password. We fine-tuned [35] the CNN model by adding a new randomly initialized classifier (output layer) and training the last fully connected layer by freezing all the layers of the base model (frozen layers represented with gray color in Figure 4) and unfreezing the last fully connected layer (unfrozen layers represented with green color in Figure 4). Freezing the layers mean that no updates to the weights are made in those layers during the training process. The new output layer of the model is then trained to take the lower level features passed through the model network and map them to the desired output classes (password), using optimization techniques, such as stochastic gradient descent (SGD) [36]. SGD is an iterative optimization algorithm, which estimates the error gradient for the CNN model during the training process and updates the weights of the model using back-propagation [37].



(**a**) 111111 (**b**) 123456

(**c**) football (**d**) passw0rd

Figure 3. Graphical representation of thermal behavior (time interval vs temperature in °C) of encryption operation using the following passwords: *111111*, *123456*, *football*, and *passw0rd*.

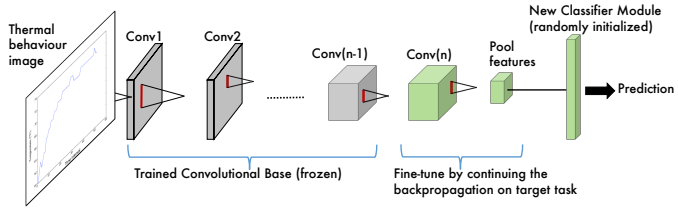


Figure 4. Network architecture used for fine-tuning.

4. ThermalAttackNet: Proposed CNN Architecture

Since most of the pre-trained CNNs come with several fully connected layers, using such a model consumes a lot of memory space on the device, as well as power. In order to overcome these challenges, we designed a CNN model, named *ThermalAttackNet*, which performs similar to popular CNNs (ResNet, VGG, NASNet, and MobileNet); however, at the same time, it consumes less power and memory comparatively. Given the fact that graphical representation of thermal behavior (as shown in Figure 3) consists of regular temperature peaks characterized by edges, we designed the CNN to be able to extract such features as accurately as possible. The architecture of ThermalAttackNet is illustrated in Figure 5. ThermalAttackNet consists of 6 convolutional layers (denoted by Conv2D in Figure 5), and we discard the fully connected layers in favor of retaining higher resolution feature maps at the deepest output layer. This also reduces the number of parameters (only 48,804) used in ThermalAttackNet compared to ResNet, VGG, NASNet, and MobileNet (as shown in Table 1). In Figure 5, it should be kept in mind that X is a variable batch size, which will depend on the implementation of the model, and C is the output classes, which is 4 (passwords) in our case. Each convolutional layer (Conv2D) performs convolution with a filter bank to produce a set of feature maps and then an element-wise rectified-

linear non-linearity (ReLU) $max(0, x)$ is applied. Following that, max-pooling (denoted as MaxPooling2D in Figure 5) is used to achieve translation invariance over small spatial shifts in the input image. Table 1 shows the comparison between ThermalAttackNet and other popular models.

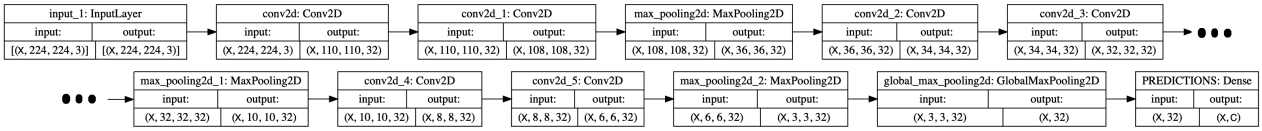


Figure 5. An illustration of the ThermalAttackNet architecture.

Note: ThermalAttackNet is trained on thermal dataset by performing augmentation to the data improve its training. The following data augmentation approaches were performed on the dataset: Horizontal and Vertical Shift, Random Zoom, and Shear Intensity.

Table 1. Comparison between models based on disk size and parameters.

	Size	Parameters
ResNet152v2	232 MB	60,380,648
NASNetMobile	23 MB	5,326,716
VGG19	549 MB	143,667,240
MobileNetv2	14 MB	3,538,984
ThermalAttackNet	0.455 MB	48,804

5. Experimental and Evaluation Results

From the 500 graphical data for each password label, we separated 100 graphical data for cross-validation testing purpose, whereas 75% of the remaining 400 graphical data for each password label were used for training, and rest of the 25% is used for validation during the training period. Validation data is used to provide an unbiased evaluation of a model fit on the training dataset while tuning hyperparameters of the model. Table 2 shows the training prediction accuracy, and Table 3 shows the testing prediction accuracy achieved by MobileNetv2, NASNetMobile, ResNetv2, VGG19, and ThermalAttackNet, respectively, on different Linux governors: conservative (cons.), ondemand (ond.), performance (perf.), interactive (inter.), and powersaver (pow.).

Table 2. Training prediction accuracy (%) achieved by different CNNs on different Linux governors: conservative (cons.), ondemand (ond.), performance (perf.).

	cons.	ond.	perf.	inter.	pow.
ResNet152v2	45.88	46.88	65.63	29	41.63
NASNetMobile	26.69	27.69	34.25	25.81	27.69
VGG19	25.44	26.19	26	24.63	26.19
MobileNetv2	55.63	66.06	**69.69**	42.56	52.75
ThermalAttackNet	25	27	25.06	25.19	25.38

Table 3. Testing prediction accuracy (%) achieved by different CNNs on different Linux governors: conservative (cons.), ondemand (ond.), performance (perf.).

	cons.	ond.	perf.	inter.	pow.
ResNet152v2	25.99	**31.999**	31	25.499	25.499
NASNetMobile	27.5	25.7499	31	29.2499	27.25
VGG19	27.75	31.4999	28.49999	25	25
MobileNetv2	30.75	25.4999	25.7499	24.5	24.75
ThermalAttackNet	25.75	26	25	25	25

5.1. Which CNN Model Is Best at Predicting Password

In Table 2, we could notice that MobileNetv2 achieves the highest training prediction accuracy of 69.6875 for performance governor; however, for the same governor, the testing prediction accuracy drops to 25.7499% (see Table 3). Since testing prediction accuracy is more important to determine if the CNN is able to predict accurately, based on Table 3, ResNet152v2 achieves the best prediction accuracy of 31.999%. Therefore, among these compared CNN models, ResNet152v2 is best at predicting password using our proposed thermal side-channel attack.

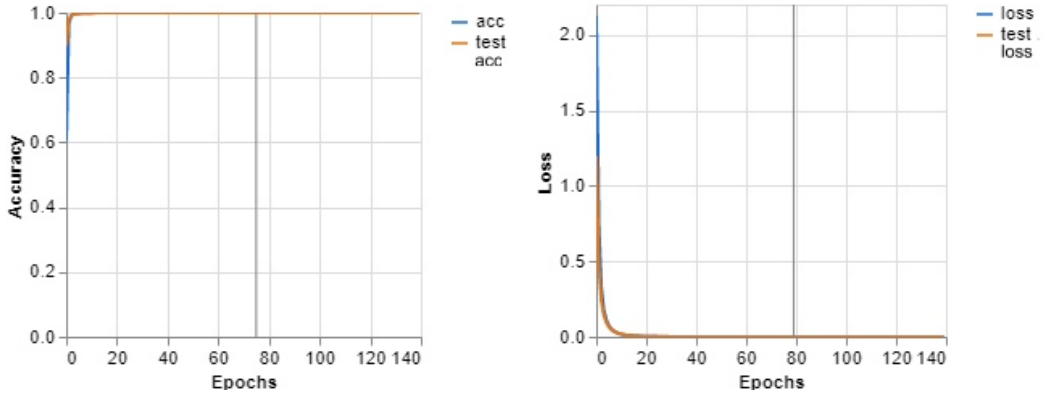
Which governor is least secure: From Table 3, it is evident that ondemand governor is the least secure among other Linux governors if ResNet152v2 is used as the model for the attack.

5.2. Power Consumption of CNNs

The average power consumption (in Watt) during inference while utilizing ResNet15v2, MobileNetv2, VGG19, NASNetMobile, and ThermalAttackNet on ondemand governor is 10.69, 9.56, 10.67, 8.79, and 7.63, respectively. Given the fact that ThermalAttackNet is fraction of a size of popular CNNs (see Table 1) while being able to predict close to other popular CNNs (see Table 3), utilizing ThermalAttackNet for such an attack on the device is more power efficient.

6. Extensive Evaluation on a Commercial Mobile Device

To evaluate the efficacy of the ThermalAttackNet on a commercial device to predict passwords via thermal side-channel, in general, we extended the evaluation by encrypting a text file, as mentioned in Section 3.2, on Exynos 5422 MPSoC, which is utilized in Samsung Galaxy devices, with 25 most commonly used passwords in 2018 [23]. To make the attack more realistic, as could be performed by an attacker or malicious program, for each password, the encryption was performed more than 400 times, and the dataset of thermal records for each password was not equal. Figure 6 shows the training accuracy (Figure 6a) and loss (Figure 6b) for 140 epoch. An epoch is a term used to indicate the number of passes of the entire training dataset the machine learning model has completed. Figure 7 shows the confusion matrix [38], which is a table that is used to describe the performance of a classifier or classification model on a set of test data for which the true values of the classes are known. In Figure 7, the primary axis represents the 25 most used passwords, and the X-axis represents the prediction performance against the same respective classes (25 most used passwords). From the confusion matrix (Figure 7), we could notice that the prediction accuracy of ThermalAttackNet is 100%, which means it was able to predict the 25 most used passwords for the encrypted texts all the time, thus proving the efficacy of such an attack by a malicious person or program.



(**a**) Training prediction accuracy (**b**) Training loss

Figure 6. Training prediction accuracy and loss of ThermalAttackNet on 25 most commonly used passwords in 2018 [23].

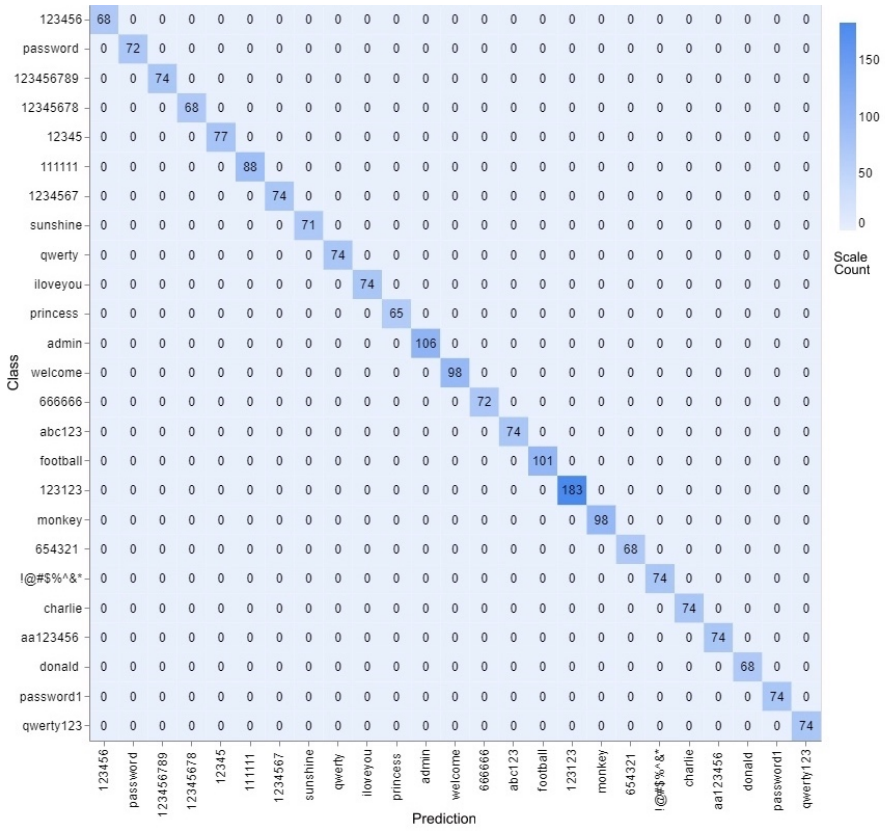


Figure 7. Confusion matrix showing the prediction performance of ThermalAttackNet for 25 most commonly used passwords in 2018 [23].

7. Discussion & Future Works

The ondemand governor is a dynamic in-kernel *cpufreq* governor that can change the CPU operating frequency depending on the CPU utilization. Here, the *cpufreq* is the subsystem of the Linux kernel that allows the operating frequency to be explicitly set on the processors. On the other hand, the performance governor sets the operating frequency of the CPUs at the highest possible frequency within a user specified range. From Table 3, we could notice that ondemand and performance governor are most vulnerable, and that is because, when the operating frequency of the CPU is set to very high, due to high power consumption, the heat dissipation on the CPU also increases significantly [2,8,9,11–14,39], which creates a peak in temperature on the CPU.

Given the fact that ondemand and performance governors are more vulnerable to attacks similar to the proposed one, some form of software/hardware mechanisms should be employed in mobile edge devices employing such governors such that either the peak temperature achieved during the encryption process could be masked or such that the peak temperature does not increase during the encryption process.

8. Conclusions

In this paper, we studied the accuracy of different CNN models, ResNet15v2, MobileNetv2, VGG19, and NASNetMobile, to predict passwords exploiting thermal side-channel attacks for different Linux governors in mobile MPSoCs. Based on empirical data, ondemand governor is the least secure among other Linux governors if ResNet152v2 is used as a CNN model for the attack. We also proposed a power-efficient CNN, ThermalAttackNet, which is able to predict passwords almost equally as ResNet152v2 CNN, however, in a more power-efficient manner, while consuming less disk storage memory on the device.

9. Code Availability

The program codes to implement the attack and generate the dataset could be accessed from https://github.com/somdipdey/ThermalAttackNet (accessed on 25 May 2021).

Author Contributions: Conceptualization, S.D.; methodology, S.D.; software, S.D.; validation, S.D.; formal analysis, S.D.; investigation, S.D.; resources, S.D.; data curation, S.D.; writing—original draft preparation, S.D.; writing—review and editing, S.D., A.K.S. and K.M.-M.; visualization, S.D.; supervision, S.D. and A.K.S.; project administration, S.D.; funding acquisition, S.D. All authors read and agreed to the published version of the manuscript.

Funding: This work is supported by Nosh Technologies under Grant nosh/agri-tech-000001 and by the U.K. Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/R02572X/1 and Grant EP/P017487/1.

Data Availability Statement: Not applicable, the study does not report any data.

Conflicts of Interest: This research was pursued such that part of the proposed methodology could be implemented to secure the commercial mobile application named nosh—Food Stock Management https://nosh.tech (accessed on 21 March 2021).

References

1. Dinh, T.Q.; Tang, J.; La, Q.D.; Quek, T.Q. Offloading in mobile edge computing: Task allocation and computational frequency scaling. *IEEE Trans. Commun.* **2017**, *65*, 3571–3584.
2. Singh, A.K.; Dey, S.; Basireddy, K.R.; McDonald-Maier, K.; Merrett, G.V.; Al-Hashimi, B.M. Dynamic Energy and Thermal Management of Multi-Core Mobile Platforms: A Survey. *IEEE Des. Test.* **2020**, *37*, 25–33. [CrossRef]
3. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge computing: Vision and challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [CrossRef]
4. Ambrose, J.A.; Ragel, R.G.; Jayasinghe, D.; Li, T.; Parameswaran, S. Side channel attacks in embedded systems: A tale of hostilities and deterrence. In Proceedings of the 2015 16th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, USA, 2–4 March 2015; pp. 452–459.
5. De Haas, J. *Side Channel Attacks and Countermeasures for Embedded Systems*; Black Hat: Las Vegas, NV, USA, 2007; p. 82.
6. Hutter, M.; Schmidt, J.M. The temperature side channel and heating fault attacks. In *International Conference on Smart Card Research and Advanced Applications*; Springer: Cham, Switzerland, 2013; pp. 219–235.
7. van Elsloo, T. Multi-Objective Optimization of Secure Embedded Systems Architectures, 2016.
8. Dey, S.; Guajardo, E.Z.; Basireddy, K.R.; Wang, X.; Singh, A.K.; McDonald-Maier, K. Edgecoolingmode: An agent based thermal management mechanism for dvfs enabled heterogeneous mpsocs. In Proceedings of the 2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID), Delhi, India, 5–9 January 2019; pp. 19–24.
9. Dey, S.; Singh, A.K.; Wang, X.; McDonald-Maier, K.D. DeadPool: Performance Deadline Based Frequency Pooling and Thermal Management Agent in DVFS Enabled MPSoCs. In Proceedings of the 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), Paris, France, 21–23 June 2019; pp. 190–195.
10. Dey, S.; Singh, A.K.; Prasad, D.K.; Mcdonald-Maier, K.D. SoCodeCNN: Program Source Code for Visual CNN Classification Using Computer Vision Methodology. *IEEE Access* **2019**, *7*, 157158–157172. [CrossRef]
11. Isuwa, S.; Dey, S.; Singh, A.K.; McDonald-Maier, K. Teem: Online thermal-and energy-efficiency management on cpu-gpu mpsocs. In Proceedings of the 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), Florence, Italy, 25–29 March 2019; pp. 438–443.
12. Dey, S.; Singh, A.; Wang, X.; McDonald-Maier, K. User Interaction Aware Reinforcement Learning for Power and Thermal Efficiency of CPU-GPU Mobile MPSoCs. In Proceedings of the 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 9–13 March 2020; pp. 1728–1733.
13. Dey, S.; Singh, A.K.; Saha, S.; Wang, X.; McDonald-Maier, K.D. RewardProfiler: A Reward Based Design Space Profiler on DVFS Enabled MPSoCs. In Proceedings of the 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), Paris, France, 21–23 June 2019; pp. 210–220.
14. Masti, R.J.; Rai, D.; Ranganathan, A.; Müller, C.; Thiele, L.; Capkun, S. Thermal Covert Channels on Multi-core Platforms. In Proceedings of the USENIX Security Symposium, Washington, DC, USA, 12–14 August 2015.
15. Bartolini, D.B.; Miedl, P.; Thiele, L. On the capacity of thermal covert channels in multicores. In Proceedings of the Eleventh European Conference on Computer Systems, London, UK, 18–21 April 2016; p. 24.
16. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [CrossRef] [PubMed]
17. Chakradhar, S.; Sankaradas, M.; Jakkula, V.; Cadambi, S. A dynamically configurable coprocessor for convolutional neural networks. *ACM SIGARCH Comput. Archit. News* **2010**, *38*, 247–257. [CrossRef]

18. Dey, S.; Kalliatakis, G.; Saha, S.; Singh, A.K.; Ehsan, S.; McDonald-Maier, K. Mat-cnn-sopc: Motionless analysis of traffic using convolutional neural networks on system-on-a-programmable-chip. In Proceedings of the 2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Edinburgh, UK, 6–9 August 2018; pp. 291–298.
19. Dey, S.; Singh, A.K.; Prasad, D.K.; Mcdonald-Maier, K.D. IRON-MAN: An Approach To Perform Temporal Motionless Analysis of Video using CNN in MPSoC. *IEEE Access* **2020**, *8*, 137101–137115. [CrossRef]
20. Kalliatakis, G.; Ehsan, S.; Fasli, M.; Leonardis, A.; Gall, J.; McDonald-Maier, K.D. Detection of Human Rights Violations in Images: Can Convolutional Neural Networks help? In Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications-Volume 5: VISAPP, Porto, Portugal, 27 February–1 March 2017.
21. Zhang, G.; Patuwo, B.E.; Hu, M.Y. Forecasting with artificial neural networks: The state of the art. *Int. J. Forecast.* **1998**, *14*, 35–62. [CrossRef]
22. The 25 Worst Passwords of 2017. Available online: http://fortune.com/2017/12/19/the-25-most-used-hackable-passwords-2017-star-wars-freedom (accessed on 31 January 2018).
23. The 25 Most Popular Passwords of 2018 Will Make You Feel Like a Security Genius. Available online: https://gizmodo.com/the-25-most-popular-passwords-of-2018-will-make-you-fee-1831052705 (accessed on 31 January 2018).
24. Rijmen, V.; Daemen, J. Advanced Encryption Standard. Available online: https://csrc.nist.gov/publications/detail/fips/197/final (accessed on 25 May 2021).
25. Odroid-XU4. Available online: https://goo.gl/KmHZRG (accessed on 23 July 2018).
26. Pallipadi, V.; Starikovskiy, A. The Ondemand Governor. Available online: https://www.kernel.org/doc/ols/2006/ols2006v2-pages-223-238.pdf (accessed on 25 May 2021).
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
28. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.
29. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet Classification With Deep Convolutional Neural Networks. Available online: https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf (accessed on 25 May 2021).
30. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [CrossRef]
31. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
32. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
33. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8697–8710.
34. Exynos 5 Octa (5422). Available online: https://www.samsung.com/exynos (accessed on 23 July 2018).
35. Lin, T.Y.; RoyChowdhury, A.; Maji, S. Bilinear cnn models for fine-grained visual recognition. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1449–1457.
36. Bottou, L. Large-scale machine learning with stochastic gradient descent. In Proceedings of the COMPSTAT'2010, Paris, France, 22–27 August 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 177–186.
37. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
38. Visa, S.; Ramsay, B.; Ralescu, A.L.; Van Der Knaap, E. Confusion Matrix-based Feature Selection. *MAICS* **2011**, *710*, 120–127.
39. Iranfar, A.; Kamal, M.; Afzali-Kusha, A.; Pedram, M.; Atienza, D. Thespot: Thermal stress-aware power and temperature management for multiprocessor systems-on-chip. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2018**, *37*, 1532–1545. [CrossRef]